CHAPTER 9

LISTING OF COMPUTER CODES

```
C     ********************************************************************
C     ********************************************************************
C     *  IMPLICIT FINITE-DIFFERENCE METHOD FOR SOLVING THE               *
C     *  PARABOLIC EQUATION : WIDE ANGLE CAPABILITY                      *
C     ********************************************************************
C     ********************************************************************
C     *  D. LEE AND G. BOTSEAS, CODE 3332                                *
C     *  NAVAL UNDERWATER SYSTEMS CENTER                                 *
C     *  NEW LONDON, CONNECTICUT 06320, U.S.A.                           *
C     ********************************************************************
C     ********************************************************************
C     *  WIDE ANGLE CONTRIBUTION BY                                      *
C     *  K. E. GILBERT                                                   *
C     *  NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY                   *
C     *  NSTL STATION, MS 39529                                          *
C     ********************************************************************
C     ********************************************************************
C     *  THIS PROGRAM IS AN IMPROVED VERSION OF THE IFD MODEL REPORTED   *
C     *  IN NUSC TECH. REPORT 6659, 27 MAY 1982, BY LEE AND BOTSEAS.     *
C     ********************************************************************
C     ********************************************************************
C     *  VAX-11/780 VERSION - FORTRAN IV+ - 22 FEB 1983                  *
C     ********************************************************************
C     *** ALPHABETICAL LIST OF PROGRAM VARIABLES FOLLOWS:                 *
C     ********************************************************************
C     *** ACOFX - COEFFICIENT 'A' AT BOTTOM - AT ADVANCED RANGE - NOT USED
C     *** ACOFY - COEFFICIENT 'A' AT BOTTOM - AT PRESENT RANGE  - NOT USED
C     *** ALPHA - VOLUME ATTENUATION - DB/METER
C     *** ATT   - ATTENUATION COEFFICIENT FOR ARTIFICIAL ABSORBING LAYER
C     *** BCOF  - COEFFICIENT 'B'  - RANGE INDEPENDENT - NOT USED
C     *** BETA  - ARRAY - ATTENUATION IN LAYERS - DB/WAVELENGTH
C     *** BOTX  - COMPLEX PRESSURE AT BOTTOM AT ADVANCED RANGE RA+DR
C     *** BOTY  - COMPLEX PRESSURE AT BOTTOM AT PRESENT RANGE RA
C     *** BTA   - ARRAY - PARTIAL SOLUTION OF SYSTEM OF EQUATIONS
C     *** C0    - REFERENCE SPEED OF SOUND - METERS/SEC
C     *** CSVP  - ARRAY - SOUND VELOCITY - METERS/SEC
C     *** DEG   - CONVERSION FACTOR - DEGREES/RADIAN
C     *** DR    - RANGE STEP - METERS
C     *** DR1   - LAST DR USED IN ROUTINE DIAG
C     *** DZ    - DEPTH INCREMENT OF SOLUTION - METERS
C     *** DZZ   - DEPTH INCREMENT FOR ADJUSTING LAYER DEPTHS IN SLOPING BOTTOM
C     *** FRQ   - FREQUENCY - HZ
C     *** HNK   - HANKEL FUNCTION H0(1)
C     *** HNKL  - EXTERNAL FUNCTION - COMPUTES HANKEL FUNCTION H0(1)
C     *** IBOT  - BOTTOM DEPTH PRINT FLAG
C                   = 0 - DO NOT PRINT BOTTOM DEPTHS
C                   = 1 - PRINT BOTTOM DEPTHS
C     *** IDIAG - DIAGONAL UPDATE FLAG
C     *** IHNK  - HANKEL FUNCTION FLAG
C                   = 0 - HANKEL FUNCTION NOT USED. 10*LOG(R) ADDED TO
C                         SOLUTION.
C                   = 1 - STARTING FIELD DIVIDED BY HANKEL FUNCTION.
C                         SOLUTION MULTIPLIED BY HANKEL FUNCTION BEFORE
C                         COMPUTING PROPAGATION LOSS.
```

```
C      *** ILYR  - INDEX FOR ARRAYS BETA, ZLYR, AND RHO
C      *** IPZ   - EVERY IPZ'TH VALUE IN DEPTH IS PRINTED
C      *** ISF   - STARTING FIELD FLAG
C                  = 0 - PROGRAM GENERATES GAUSSIAN STARTING FIELD
C                        AT RANGE = 0.0. SEE SUBROUTINE SFIELD.
C                  = 1 - USER SUPPLIES STARTING FIELD. SEE SUBROUTINE
C                        UFIELD.
C      *** ISFLD - STARTING FIELD PRINT FLAG
C                  = 0 - DO NOT PRINT STARTING FIELD
C                  = 1 - PRINT STARTING FIELD
C      *** ISVP  - SVP PRINT FLAG
C                  = 0 - DO NOT PRINT SOUND VELOCITY PROFILE
C                  = 1 - PRINT SOUND VELOCITY PROFILE
C      *** ITEMP - TEMPORARY VARIABLE
C      *** ITRK  - INDEX FOR ARRAY TRACK
C      *** ITYPEB- TYPE OF BOTTOM
C              0 - RIGID BOTTOM - PROGRAM SUPPLIES BOTTOM CONDITION
C              1 - NOT RIGID    - USER SUPPLIES BOTTOM CONDITION
C                                 - SEE SUBROUTINE BCON
C              2 - NOT RIGID    - ABSORBING LAYER INTRODUCED
C                                 - FOLLOWS CONTOUR OF BOTTOM
C              3 - NOT RIGID    - ABSORBING LAYER INTRODUCED - BUT
C                                 - BOTTOM OF ABSORBING LAYER KEPT FLAT
C      *** ITYPES- TYPE OF SURFACE
C                  = 0 - PRESSURE RELEASE. SCON SETS SURY AND SURX = 0.0
C                  = .NE. 0 - USER INSERTS CODE IN SCON TO COMPUTE SURY AND SURX
C      *** IWZ   - INDEX INCREMENT OF RECEIVER SOLUTIONS TO BE WRITTEN ON DISK
C      *** IXSVP - ARRAY OF POINTERS WHICH POINT TO ENTRIES IN CSVP AND ZSVP
C                - IXSVP(1) POINTS TO BOTTOM DEPTH AND SPEED IN LAYER 1
C                - IXSVP(2) POINTS TO BOTTOM DEPTH AND SPEED IN LAYER 2
C                  ETC.
C      *** KSVP  - SVP PROFILE FLAG
C            0   - PROFILE IS ON CARDS - SEE SUBROUTINE SVP
C            1   - USER SUPPLIES PROFILE 1 - SEE SUBROUTINE USVP
C            2   - USER SUPPLIES PROFILE 2 - SEE SUBROUTINE USVP
C            .     .     .     .     .     .
C            N   - USER SUPPLIES PROFILE N - SEE SUBROUTINE USVP
C      *** MLYR  - TEMPORARY - NUMBER OF LAYERS INVOLVED IN SPECIFIC CALCULATION
C      *** MM    - INDEX - MM+1 POINTS TO FIRST VALUE OF ARTIFICIAL ABSORBING
C                  LAYER IN ARRAY U
C      *** MXLYR - PARAMETER - MAXIMUM NUMBER OF LAYERS
C                            - MAX DIMENSION OF ARRAYS BETA,RHO,ZLYR,IXSVP
C      *** MXN   - PARAMETER - MAXIMUM DIMENSION OF C, X, Y, R12 AND U ARRAYS
C      *** MXSVP - PARAMETER - MAXIMUM DIMENSION OF ARRAYS CSVP AND ZSVP
C      *** MXTRK - PARAMETER - MAXIMUM DIMENSION OF ARRAY TRACK
C      *** N     - NUMBER OF EQUI-SPACED POINTS IN U
C                - INCLUDES BOTTOM POINT - DOES NOT INCLUDE SURFACE POINT
C      *** N1    - DIAGONAL ELEMENTS N1 THRU N WILL BE COMPUTED
C      *** NA    - NUMBER OF POINTS IN ABSORBING LAYER
C      *** NIU   - PARAMETER - INPUT UNIT NUMBER
C      *** NLYR  - NUMBER OF LAYERS
C      *** NOLD  - NUMBER OF RECEIVER DEPTHS ON ENTRY TO ROUTINE CRNK
C      *** NOU   - PARAMETER - OUTPUT UNIT NUMBER
C      *** NPU   - PARAMETER - PRINTER UNIT NUMBER
C      *** NSVP  - NUMBER OF POINTS IN CSVP AND ZSVP
C      *** NWSVP - NUMBER OF POINTS IN LAYER 1 SVP
C      *** NZ    - NUMBER OF SOLUTION DEPTHS TO BE WRITTEN ON DISK
C      *** OLDR  - RANGE INCREMENT AT START OF PROBLEM
C      *** PDR   - RANGE INCREMENT AT WHICH SOLUTION IS PRINTED - METERS
C      *** PDZ   - DEPTH INCREMENT AT WHICH SOLUTION IS PRINTED - METERS
C      *** PI    - THE VALUE OF PI
C      *** PL    - PROPAGATION LOSS - DB
C      *** R1    -      RANGE AT WHICH BOTTOM DEPTH IS AVAILABLE - METERS
```

```
C      *** R2     - NEXT RANGE AT WHICH BOTTOM DEPTH IS AVAILABLE - METERS
C      *** R12    - ARRAY OF DENSITY RATIOS
C      *** RA     - HORIZONTAL RANGE OF STARTING FIELD FROM SOURCE - METERS
C                 - RA IS SET TO 0.0 IF STARTING FIELD IS GAUSSIAN. RA IS
C                 - INCREMENTED AS SOLUTION IS MARCHED OUT IN RANGE.
C      *** RA+DR  - RANGE TO WHICH SOLUTION IS TO BE ADVANCED - METERS
C      *** RHO    - ARRAY - DENSITY IN LAYER
C      *** RMAX   - MAXIMUM RANGE OF SOLUTION - METERS
C      *** RSVP   - NEXT RANGE AT WHICH SVP IS AVAILABLE - METERS
C      *** SURX   - COMPLEX PRESSURE AT SURFACE AT ADVANCED RANGE RA+DR
C      *** SURY   - COMPLEX PRESSURE AT SURFACE AT PRESENT RANGE RA
C      *** TEMP   - TEMPORARY VARIABLE - COMPLEX
C      *** THETA  - SLOPE OF BOTTOM - RADIANS
C                   .EQ.0 -- FLAT BOTTOM
C                   .GT.0 -- SHALLOW TO DEEP
C                   .LT.0 -- DEEP TO SHALLOW
C      *** TM     - ARRAY - TIME OF DAY
C      *** TRACK  - 2 DIM. ARRAY - RANGE AND DEPTH OF WATER - METERS
C      *** U      - ARRAY - COMPLEX ACOUSTIC PRESSURE FIELD
C      *** W      - COMPLEX VARIABLE - W=W2/W2S
C      *** W1     - COMPLEX VARIABLE - TEMPORARY
C      *** W1S    - COMPLEX VARIABLE - CONJUGATE OF W1.
C      *** W2     - COMPLEX VARIABLE - TEMPORARY
C      *** W2S    - COMPLEX VARIABLE - CONJUGATE OF W2.
C      *** WA-WD  - COEFFICIENTS FOR RATIONAL FUNCTION APPROXIMATION FOR
C                   SQUARE ROOT OPERATOR.
C                   FOR TAPPERT:   WA=1, WB=.5 , WC=1, WD=0
C                                  PROPAGATION ANGLES UP TO APPROX. 15 DEGREES
C                   FOR CLAERBOUT: WA=1, WB=.75, WC=1, WD=.25
C                                  PROPAGATION ANGLES UP TO APPROX. 40 DEGREES
C      *** WDR    - RANGE STEP AT WHICH SOLUTION IS WRITTEN ON DISK - METERS
C      *** WDZ    - DEPTH INCREMENT AT WHICH SOLUTION IS WRITTEN ON DISK - METERS
C                   WDZ SHOULD BE SELECTED SO THAT PLOT PROGRAM DOES NOT
C                   INTERPOLATE BETWEEN WIDELY SPACED RECEIVERS.
C      *** X      - ARRAY - MAIN DIAGONAL OF MATRIX AT ADVANCED RANGE
C      *** XK0    - REFERENCE WAVE NUMBER
C      *** XPR    - RANGE AT WHICH SOLUTION IS PRINTED - METERS
C      *** XWR    - RANGE AT WHICH SOLUTION IS WRITTEN ON DISK - METERS
C      *** Y      - ARRAY - MAIN DIAGONAL OF MATRIX AT PRESENT RANGE
C      *** Z1     - DEPTH OF WATER AT RANGE R1 - METERS
C      *** Z2     - DEPTH OF WATER AT RANGE R2 - METERS
C      *** ZA     - DEPTH OF FIELD AT RANGE RA - METERS
C                 - INITIAL DEPTH OF STARTING FIELD AT RANGE RA IS
C                   AS FOLLOWS:
C                 - IF ITYPEB = 0 OR 1, ZA IS MAXIMUM DEPTH OF
C                   BOTTOM-MOST SEDIMENT LAYER AT INITIAL RANGE OF
C                   STARTING FIELD. IF ITYPEB = 2 OR 3, ZA IS MAXIMUM
C                   DEPTH OF ARTIFICIAL ABSORBING LAYER AT INITIAL
C                   RANGE OF STARTING FIELD. PROGRAM INSERTS LAYER.
C                   RHO AND BETA ARE OBTAINED FROM LAYER ABOVE.
C                   SPEED IS BOTTOM-MOST SPEED FROM LAYER ABOVE.
C                   AS SOLUTION PROGRESSES,
C                   ZA IS UPDATED IF OCEAN BOTTOM NOT FLAT. IF ITYPEB = 3,
C                   BOTTOM OF ABSORBING LAYER REMAINS FLAT.
C      *** ZLYR   - ARRAY - DEPTH TO BOTTOM OF LAYER - METERS
C      *** ZI     - DEPTH OF RECEIVER 'I' - METERS
C      *** ZS     - SOURCE DEPTH - METERS
C      *** ZSVP   - ARRAY - DEPTH OF SOUND VELOCITY - METERS
C      ******************************************************************
C      *** INPUT
C      ******************************************************************
C          INPUT UNIT NUMBER = NIU
C          INPUT FILE NAME   = IFD.IN
```

```
C       CONTENTS:  CARD IMAGES IN FREE FORMAT
C       CARD 1   :  FRQ,ZS,C0,ISF,RA,ZA,N,IHNK,ITYPEB,ITYPES
C       CARD 2   :  RMAX,DR,WDR,WDZ,PDR,PDZ,ISFLD,ISVP,IBOT
C       CARD 2A  :  WA,WB,WC,WD
C       CARD 3   :  R1,Z1   **
C       CARD 4   :  R2,Z2     *     BOTTOM PROFILE
C         .           .       **    RANGE, WATER DEPTH (METERS)
C         .           .       *
C       CARD N   :    .      **
C       CARD N+1:   -1,-1
C       CARD N+2:   RSVP                        *****************
C       CARD N+3:   KSVP                                        *
C       CARD N+4:   NLYR                                        *
C       CARD N+5:   ZLYR(I),RHO(I),BETA(I)   **                 *
C       CARD N+6:   ZSVP(1),CSVP(1)            *                **  REPEAT
C       CARD N+7:   ZSVP(2),CSVP(2)            **  REPEAT       *** FOR EACH
C         .           .       .                *** FOR EACH     **  PROFILE
C         .           .       .                **  LAYER        *
C         .           .       .                *                *
C       CARD N+M:   ZSVP(J),CSVP(J)           **                *
C                                             ****************
C
C     ******************************************************************
C     *** QUICK REFERENCE AND NOTES FOR CARD INPUT
C     *** UNITS: METERS AND METERS/SEC EXCEPT AS NOTED
C     ******************************************************************
C         FRQ = FREQUENCY (HZ)
C         ZS  = SOURCE DEPTH
C         C0  = REFERENCE SOUND SPEED. IF C0 = 0.0, C0 IS SET TO AVERAGE
C               SPEED IN FIRST LAYER.
C         ISF = STARTING FIELD FLAG. 0 = GAUSSIAN. 1 = USER FIELD.
C               IF ISF = 0, RA IS SET TO ZERO.
C         RA  = HORIZONTAL RANGE FROM SOURCE TO STARTING FIELD.
C               RA IS SET TO 0.0 IF ISF = 0.
C         ZA  = DEPTH OF STARTING FIELD AT RANGE RA. IF ZA = 0.0, ZA IS SET
C               TO MAX DEPTH OF BOTTOM LAYER IN FIRST PROFILE. IF ITYPEB =
C               2 OR 3 AND ZA = 0.0, ZA IS SET TO (4/3)*MAX DEPTH OF BOTTOM
C               LAYER. IF ITYPEB = 2 OR 3 AND ZA NOT ZERO, THE ARTIFICIAL
C               BOTTOM LAYER IS EXTENDED TO ZA METERS PROVIDED THAT ZA
C               IS GREATER THAN OR EQUAL TO MAX DEPTH OF BOTTOM LAYER
C               IN FIRST PROFILE.
C         N   = NUMBER OF EQUISPACED RECEIVERS IN STARTING FIELD. IF N = 0,
C               N IS SET SO THAT THE RECEIVER DEPTH INCREMENT IS LESS THAN
C               OR EQUAL TO 1/4 WAVELENGTH. IF N IS GREATER THAN MXN,
C               N IS SET TO MXN.
C       IHNK  = HANKEL FUNCTION FLAG. IHNK = 0, DON'T USE HANKEL FUNCTION.
C               IHNK = 1, DIVIDE STARTING FIELD BY HANKEL FUNCTION, THEN
C               MULTIPLY THE SOLUTION FIELD BY HANKEL FUNCTION BEFORE
C               COMPUTING PROPAGATION LOSS. IF STARTING FIELD IS GAUSSIAN,
C               IHNK SHOULD BE SET TO 0. IF STARTING FIELD IS ELLIPTIC,
C               IHNK SHOULD BE SET TO 1.
C      ITYPEB = TYPE OF BOTTOM PROCESSING.
C             = 0 - RIGID BOTTOM. PROGRAM SUPPLIES BOTTOM CONDITION.
C             = 1 - USER SUPPLIES BOTTOM CONDITION. USER WRITES SUBROUTINE
C                   BCON.
C             = 2 - ARTIFICIAL ABSORBING BOTTOM INTRODUCED. FOLLOWS CONTOUR
C                   OF WATER/BOTTOM INTERFACE.
C             = 3 - ARTIFICIAL ABSORBING BOTTOM INTRODUCED. BOTTOM OF
C                   LAYER KEPT FLAT.
C      ITYPES = TYPE OF SURFACE
```

```
C              =   0 - PRESSURE RELEASE. SCON SETS SURY AND SURX = 0.0
C             NOT 0 - USER INSERTS CODE IN SCON TO COMPUTE SURY AND SURX
C
C
C        RMAX = MAXIMUM RANGE OF SOLUTION
C        DR   = RANGE STEP. IF DR = 0, DR IS SET TO 1/2 WAVELENGTH.
C               IF BOTTOM OF PROBLEM IS NOT FLAT, DR IS RECOMPUTED
C               SO THAT MAX DEPTH IS EITHER INCREMENTED OR DECREMENTED
C               BY DZ. SOLUTION IS COMPUTED EVERY DR METERS.
C        WDR  = RANGE STEP AT WHICH SOLUTION IS WRITTEN ON DISK. IF WDR NOT 0,
C               AN OUTPUT DISK FILE IS ASSIGNED.
C               ROUNDED TO NEAREST DR.
C        WDZ  = DEPTH INCREMENT AT WHICH SOLUTION IS WRITTEN ON DISK.
C               ROUNDED TO NEAREST DZ.
C        PDR  = RANGE STEP AT WHICH SOLUTION IS PRINTED.
C               ROUNDED TO NEAREST DR.
C        PDZ  = DEPTH INCREMENT AT WHICH SOLUTION IS PRINTED.
C               ROUNDED TO NEAREST DZ.
C      ISFLD = 0 - DON'T PRINT STARTING FIELD.
C            = 1 - PRINT STARTING FIELD.
C       ISVP = 0 - DON'T PRINT SOUND VELOCITY PROFILE.
C            = 1 - PRINT SOUND VELOCITY PROFILE.
C       IBOT = 0 - DON'T PRINT BOTTOM DEPTHS.
C            = 1 - PRINT BOTTOM DEPTHS.
C
C      WA - WD    - COEFFICIENTS FOR RATIONAL FUNCTION APPROXIMATION FOR
C                   SQUARE ROOT OPERATOR.
C                   FOR TAPPERT:    WA=1, WB=.5 , WC=1, WD=0
C                                   PROPAGATION ANGLES UP TO APPROX. 15 DEGREES
C                   FOR CLAERBOUT: WA=1, WB=.75, WC=1, WD=.25
C                                   PROPAGATION ANGLES UP TO APPROX. 40 DEGREES
C
C        R1,Z1 = BOTTOM PROFILE. FIRST RANGE AND DEPTH OF WATER.
C        R2,Z2 = ETC. (-1,-1) MARKS THE END OF THE BOTTOM PROFILE.
C        RSVP  = RANGE OF SVP
C        KSVP  = SVP FLAG.
C              = 0 - SVP IN INPUT RUNSTREAM
C              = NOT ZERO - PROFILE (CARDS N+4 THRU N+M) IS SUPPLIED BY
C                USER. USER WRITES SUBROUTINE USVP. KSVP MAY BE USED IN
C                COMPUTED GOTO STATEMENT TO TRANSFER CONTROL IN USVP.
C        NLYR  = NUMBER OF LAYERS. IF ITYPEB = 2 OR 3, PROGRAM INSERTS
C                AN ARTIFICIAL LAYER AND INCREMENTS NLYR BY 1.
C      ZLYR(I) = MAX DEPTH OF LAYER I IN PROFILE
C      RHO(I)  = DENSITY IN LAYER I (G/CM**3)
C      BETA(I) = ATTENUATION IN LAYER I (DB/WAVELENGTH)
C                IF BETA(I) IS NEGATIVE, ATTENUATION IS COMPUTED.
C      ZSVP    = DEPTH ARRAY FOR SOUND SPEED
C      CSVP    = SPEED ARRAY FOR SOUND SPEED
C      ZSVP(1) = DEPTH OF WATER TO TOP OF LAYER I
C      CSVP(1) = SPEED OF SOUND AT TOP OF LAYER I
C      ZSVP(J) = DEPTH OF WATER TO BOTTOM OF LAYER I
C      CSVP(J) = SPEED OF SOUND AT BOTTOM OF LAYER I
C                IF ONLY ONE SVP INPUTTED, IT IS USED THRU ENTIRE PROBLEM.
C                IF MORE THAN ONE SVP INPUTTED, LAST SVP IS USED THRU REMAINDER
C                OF PROBLEM.
C
C     ******************************************************************
C     *** OUTPUT
C     ******************************************************************
C         OUTPUT UNIT NUMBER = NOU
```

```
C          OUTPUT FILE NAME    = IFD.OUT
C          CONTENTS:  AS FOLLOWS - UNFORMATTED
C
C            FRQ,ZS,C0,ISF,RA,ZA,N,IHNK,ITYPEB,ITYPES,RMAX,DR,WDR,DZ,NLYR,ZLYR,
C            RHO,BETA
C            NZ,RA,WDZ,(U(I),I=IWZ,N,IWZ) - (FOR EACH WRITE RANGE WDR)
C
C          PRINTER UNIT NUMBER = NPU
C     ******************************************************************
C
      PARAMETER MXLYR=101,MXN=10000,MXSVP=101,MXTRK=101,NIU=1,
     C           NOU=2,NPU=6
      COMPLEX ACOFX,ACOFY,BCOF,BOTX,BOTY,BTA,HNK,HNKL,SURX,SURY,TEMP,
     C         U,X,Y
      COMMON /IFDCOM/ACOFX,ACOFY,ALPHA,BCOF,BETA(MXLYR),BOTX,BOTY,
     C        BTA(MXN),C0,CSVP(MXSVP),DR,DR1,DZ,FRQ,IHNK,ISF,ITYPEB,
     C        ITYPES,IXSVP(MXLYR),KSVP,N,N1,NLYR,NSVP,NWSVP,R12(MXN),RA,
     C        RHO(MXLYR),RSVP,SURX,SURY,THETA,TRACK(MXTRK,2),U(MXN),
     C        X(MXN),XK0,Y(MXN),ZA,ZLYR(MXLYR),ZP,ZS,ZSVP(MXSVP)
C
      COMPLEX W,W1,W1S,W2,W2S,XN2
      COMMON /WIDANG/W,WA,WB,WC,WD,W1,W1S,W2,W2S,XN2
      DATA PI/3.141592654/,DEG/57.29578/
      BYTE TM(8)
C
C     *** TIME OF DAY AT START OF RUN
      CALL TIME(TM)
10    FORMAT(5X,8A1/)
C
C     *** READ INPUT PARAMETERS
      CALL ASSIGN(NIU,'IFD.IN')
      READ(NIU,*) FRQ,ZS,C0,ISF,RA,ZA,N,IHNK,ITYPEB,ITYPES
      READ(NIU,*) RMAX,DR,WDR,WDZ,PDR,PDZ,ISFLD,ISVP,IBOT
      READ(NIU,*) WA,WB,WC,WD
C
C     *** IF GAUSSIAN STARTING FIELD, RA MUST BE 0.
      IF(ISF.EQ.0) RA=0.0
C
C     *** READ BOTTOM PROFILE - RANGE,DEPTH
      DO 22 I=1,MXTRK
      READ(NIU,*) TRACK(I,1),TRACK(I,2)
      ITRK=I
C     *** END OF PROFILE?
      IF(TRACK(I,1).LT.0.0) GO TO 23
C     *** NO
22    CONTINUE
C     *** ERROR?
23    IF(ITRK.EQ.1.OR.ITRK.GT.MXTRK) GO TO 28
C     *** NO. EXTEND LAST DEPTH BEYOND MAX RANGE.
      TRACK(ITRK,1)=1.0E+38
      TRACK(ITRK,2)=TRACK(ITRK-1,2)
      ITRK=1
      R2=TRACK(ITRK,1)
      Z2=TRACK(ITRK,2)
C     *** FIND BOTTOM SEGMENT WHICH CONTAINS STARTING RANGE
25    R1=R2
      Z1=Z2
      ITRK=ITRK+1
      IF(ITRK.GT.MXTRK) GO TO 28
      R2=TRACK(ITRK,1)
```

```
      Z2=TRACK(ITRK,2)
C     *** ADVANCE TRACK IF NECESSARY SO THAT R1.LE.RA.LT.R2
      IF(RA.GE.R2) GO TO 25
C     *** R1 MUST BE LE RA
      IF(R1.LE.RA) GO TO 30
      WRITE(NPU,27)
27    FORMAT(1X,'DEPTH OF BOTTOM AT INITIAL RANGE MISSING')
      STOP
28    CONTINUE
      ITEMP=MXTRK
      WRITE(NPU,29) ITEMP
29    FORMAT(1X,'ERROR. BOTTOM PROFILE MISSING OR TOO MANY POINTS.
     CMAX IS ',I5)
      STOP
30    CONTINUE
C
C     *** COMPUTE SLOPE OF BOTTOM
      THETA=ATAN2(Z2-Z1,R2-R1)
C
C     *** READ RANGE OF SVP
      READ(NIU,*,END=33) RSVP
C     *** SVP BEYOND START RANGE?
      IF(RSVP.GT.RA) GO TO 33
C     *** NO. DETERMINE IF SVP IN RUNSTREAM OR SUPPLIED BY SUBROUTINE USVP
32    READ(NIU,*,END=33) KSVP
      IF(KSVP.EQ.0)CALL SVP(NLYR,ZLYR,RHO,BETA,IXSVP,NSVP,ZSVP,CSVP)
      IF(KSVP.NE.0)CALL USVP
C     *** ERROR IN SVP?
      IF(NSVP.NE.0) GO TO 35
C     *** YES.
33    WRITE(NPU,34) RA
34    FORMAT(1X,'SVP MISSING OR INPUT ERROR. RANGE = ',F8.1,' M.')
      STOP
C
35    CONTINUE
C     *** SAVE POINTER TO LAST SVP VALUE IN FIRST LAYER.
      NWSVP=IXSVP(1)
C
C     *** IF C0 NOT SPECIFIED, SET C0 TO AVERAGE SPEED OF FIRST PROFILE
C     *** IN FIRST LAYER AT INITIAL RANGE
      IF(C0.NE.0.0) GO TO 45
      DO 40 I=2,NWSVP
      C0=C0+(ZSVP(I)-ZSVP(I-1))*(CSVP(I-1)+.5*(CSVP(I)-CSVP(I-1)))
40    CONTINUE
      C0=C0/ZSVP(NWSVP)
45    CONTINUE
C
C     *** COMPUTE REFERENCE WAVE NUMBER
      XK0=2.0*PI*FRQ/C0
C
C     *** COMPUTE ATTENUATION - SACLANT MEMO SM-121 (JENSEN + FERLA)
C     *** MODIFIED AS FOLLOWS:
C     *** IF INPUTTED BETA IS LT 0.0, ALPHA IS COMPUTED IN DB/METER
C         AND USED FOR BETA
      ALPHA=FRQ*FRQ*(.007+(.155*1.7)/(1.7*1.7+FRQ*FRQ*.000001))*1.0E-09
C
C     *** ADJUST LAYER DEPTHS IN CASE BOTTOM SLOPES AND RA.NE.RSVP
C     *** ASSUMES LAYERS ARE PARALLEL AND FOLLOW BOTTOM CONTOUR.
C     *** LAYER DEPTHS ARE ENTERED WITH SVP.
      DO 50 ILYR=1,NLYR
```

```
50    ZLYR(ILYR)=ZLYR(ILYR)+(RA-RSVP)*TAN(THETA)
C
C     *** GET RANGE OF NEXT SVP
      READ(NIU,*,END=55) RSVP
      GO TO 56
C     *** ONLY ONE SVP - SET RSVP LARGE SO SAME SVP USED FOR WHOLE PROBLEM
55    RSVP=1.0E+38
56    CONTINUE
C
C     *** IF STARTING FIELD IS BEYOND NEXT SVP, GO BACK AND GET NEXT SVP
      IF(RSVP.LE.RA) GO TO 32
C
C     *** IF ITYPEB = 2 OR 3, AND ZA = 0, EXTEND BOTTOM BY SETTING ZA TO
C     *** 4/3 MAX DEPTH
C     *** SEE NORDA TECH NOTE 12, JAN 78, H. K. BROCK
C     *** IF ITYPEB = 2 OR 3 AND ZA NOT 0, EXTEND BOTTOM TO ZA METERS
      IF(ITYPEB.LT.2) GO TO 60
      IF(ITYPEB.GT.3) GO TO 60
C     *** EXTEND BOTTOM TO ZA METERS IF ZA NOT ZERO
C     *** EXTEND BOTTOM 4/3 MAX DEPTH IF ZA = ZERO
      IF(ZA.EQ.0.0) ZA=4.0*ZLYR(NLYR)/3.0
      IF(ZA.GE.ZLYR(NLYR)) GO TO 58
C     *** USER ATTEMPTED TO EXTEND DEPTH IN NEGATIVE DIRECTION
      WRITE(NPU,57)
57    FORMAT(1X,'ERROR. ZA RESET TO MAX DEPTH OF BOTTOM LAYER.')
      ZA=ZLYR(NLYR)
C     *** INSERT PARAMETERS FOR EXTENDED BOTTOM IN APPROPRIATE ARRAYS
58    NLYR=NLYR+1
C     *** STORE DEPTH OF ARTIFICIAL ABSORBING LAYER
      ZLYR(NLYR)=ZA
C     *** USE SAME DENSITY AND ATTENUATION AS LAYER ABOVE
      RHO(NLYR)=RHO(NLYR-1)
      BETA(NLYR)=BETA(NLYR-1)
C     *** USE BOTTOM SPEED OF LAYER ABOVE
      NSVP=NSVP+1
      IXSVP(NLYR)=NSVP
      ZSVP(NSVP)=ZLYR(NLYR)
      CSVP(NSVP)=CSVP(NSVP-1)
      GO TO 62
60    CONTINUE
C     *** IF BOTTOM NOT EXTENDED AND ZA=0, SET ZA TO MAX DEPTH INPUTTED
C     *** WITH FIRST SVP
      IF(ZA.EQ.0.0) ZA=ZLYR(NLYR)
C
C     *** IF N NOT SPECIFIED, SET N SO THAT DZ .LE. 1/4 WAVELENGTH
62    IF(N.EQ.0) N=(4*ZA*FRQ/C0)+1
      IF(N.LE.MXN) GO TO 65
      ITEMP=MXN
      WRITE(NPU,64) ITEMP
64    FORMAT(' ERROR. N TOO LARGE. N RESET TO ',I4,'.')
      N=MXN
65    CONTINUE
C
C     *** COMPUTE RECEIVER DEPTH INCREMENT - DZ MAY BE SUCH THAT RECEIVERS DO
C     *** NOT LIE EXACTLY ON LAYER INTERFACES
      DZ=ZA/N
C
C     *** IF BOTTOM IS FLAT, RANGE STEP MAY BE SPECIFIED
C     *** IF RANGE STEP NOT SPECIFIED, SET IT EQUAL TO 1/2 WAVELENGTH   !???
      IF(DR.EQ.0.0) DR=.5*C0/FRQ
```

```
C     *** IF BOTTOM IS NOT FLAT, COMPUTE RANGE STEP
C     *** IT MAY BE NECESSARY TO REDUCE DZ SUCH THAT DR IS SMALL ENOUGH
C     *** TO PRODUCE ACCURATE RESULTS
      IF(ITYPEB.NE.3.AND.THETA.NE.0.0) DR=ABS(DZ/TAN(THETA))
C     *** COMPUTE DEPTH INCREMENT FOR ADJUSTING LAYERS
      DZZ=DR*TAN(THETA)
C
C     *** GET STARTING FIELD
      IF(ISF.EQ.0) CALL SFIELD(FRQ,C0,ZS,N,DZ,U)
      IF(ISF.NE.0) CALL UFIELD
      IF(IHNK.EQ.0) GO TO 69
C
C     *** DIVIDE STARTING FIELD BY HANKEL FUNCTION IF REQUESTED BY USER
      HNK=HNKL(XK0*RA)
      DO 68 I=1,N
      U(I)=U(I)/HNK
68    CONTINUE
69    CONTINUE
C
C     *** COMPUTE DEPTH WRITE INCREMENT TO NEAREST DZ
      IF(WDZ.LT.DZ)WDZ=DZ
      IWZ=WDZ/DZ+.5
      WDZ=IWZ*DZ
C
C     *** COMPUTE DEPTH PRINT INCREMENT TO NEAREST DZ
      IPZ=PDZ/DZ+.5
      IF(PDZ.GT.0.0.AND.IPZ.EQ.0) IPZ=1
      PDZ=IPZ*DZ
C
C     *** PRINT PROBLEM PARAMETERS
      WRITE(NPU,72)
72    FORMAT(//1X,'IFD SOLUTION')
      IF(ISF.EQ.0) WRITE(NPU,73)
73    FORMAT(1X,'GAUSSIAN STARTING FIELD')
      IF(ISF.NE.0) WRITE(NPU,74)
74    FORMAT(1X,'USER STARTING FIELD')
      IF(IHNK.NE.0) WRITE(NPU,75)
75    FORMAT(1X,'STARTING FIELD DIVIDED BY HANKEL FUNCTION')
      IF(ITYPES.NE.0) WRITE(NPU,76)
76    FORMAT(1X,'USER SURFACE CONDITION')
      IF(ITYPEB.EQ.1) WRITE(NPU,77)
77    FORMAT(1X,'USER BOTTOM CONDITION')
      WRITE(NPU,80) FRQ,ZS,C0,ISF,RA,ZA,N,IHNK,ITYPEB,ITYPES
80    FORMAT(1X,'FRQ    = ',F8.2,' HZ',/,1X,
     C'ZS     = ',F8.2,' M',/,1X,
     C'C0     = ',F8.2,' M/SEC',/,1X,
     C'ISF    = ',I5,/,1X,
     C'RA     = ',F8.2,' M',/,1X,
     C'ZA     = ',F8.2,' M',/,1X,
     C'N      = ',I5,/,1X,
     C'IHNK   = ',I5,/,1X,
     C'ITYPEB = ',I5,/,1X,
     C'ITYPES = ',I5)
      WRITE(NPU,81) DR,WDR,PDR,DZ,WDZ,PDZ,WA,WB,WC,WD,RMAX
81    FORMAT(1X,'DR     = ',F8.2,' M',/,1X,
     C'WDR    = ',F8.2,' M',/,1X,
     C'PDR    = ',F8.2,' M',/,1X,
     C'DZ     = ',F8.2,' M',/,1X,
     C'WDZ    = ',F8.2,' M',/,1X,
     C'PDZ    = ',F8.2,' M',/,1X,
```

```
      C'A      = ',F8.4,' M',/,1X,
      C'B      = ',F8.4,' M',/,1X,
      C'C      = ',F8.4,' M',/,1X,
      C'D      = ',F8.4,' M',/,1X,
      C'RMAX   =',F9.1,' M',/,/,
      C2X,'LAYER  MAX DEPTH(M)       DENSITY(G/CM**3)  ATT(DB/WL)',/)
       DO 85 ILYR   =1,NLYR
       WRITE(NPU,84)ILYR,ZLYR(ILYR),RHO(ILYR),BETA(ILYR)
84     FORMAT(2X,I3,3X,E13.6,5X,E13.6,5X,E13.6)
85     CONTINUE
C
C      *** PRINT BOTTOM DEPTHS IF REQUESTED
       IF(IBOT.EQ.0) GO TO 86
       TH=THETA*DEG
       WRITE(NPU,123) R1,Z1,R2,Z2,TH
86     IF(ISFLD.EQ.0) GO TO 92
C
C      *** PRINT STARTING FIELD
       WRITE(NPU,87)
87     FORMAT(/,1X,'STARTING FIELD')
       DO 90 I=1,N
       ZI=I*DZ
       IF(U(I).NE.0.0)WRITE(NPU,89) I,ZI,U(I)
89     FORMAT(1X,I4,3X,F10.2,3X,'(',E12.5,2X,E12.5,' )')
90     CONTINUE
92     CONTINUE
C
       IF(ISVP.EQ.0) GO TO 96
C      *** PRINT SVP
       WRITE(NPU,93) RA
93     FORMAT(//1X,'SOUND VELOCITY PROFILE AT RANGE ',F10.1,' METERS',/)
       ILYR=1    !!!!!!!!!!!!!!!!
       DO 95 I=1,NSVP
       WRITE(NPU,94) I,ZSVP(I),CSVP(I)
94     FORMAT(1X,I4,3X,F8.1,3X,F8.1)
C       IF(I.NE.IXSVP(ILYR)) GO TO 95 !!!!!!!!!!!!!!
C       WRITE(NPU,84)ILYR,ZLYR(ILYR),RHO(ILYR),BETA(ILYR)!!!!!!!!!!!!!!!!!!!!
       ILYR=ILYR+1
95     CONTINUE
96     CONTINUE
C
C      *** COMPUTE 'B' COEFFICIENT
       BCOF=CMPLX(0.0,.5/XK0)
C       BCOF=CMPLX(0.0,1.0/XK0)*(WA/WC)*(WB/WA-WD/WC)
C
C      *** ASSIGN OUTPUT FILE IF OUTPUT REQUESTED
       IF(WDR.EQ.0.0)GO TO 98
       CALL ASSIGN(NOU,'IFD.OUT')
C
C      *** WRITE SELECTED PARAMETERS FOLLOWED BY STARTING FIELD
       WRITE(NOU)FRQ,ZS,C0,ISF,RA,ZA,N,IHNK,ITYPEB,ITYPES,RMAX,DR,WDR,DZ,
      CNLYR,ZLYR,RHO,BETA
       NZ=N/IWZ
       WRITE(NOU) NZ,RA,WDZ,(U(I),I=IWZ,N,IWZ)
C
98     CONTINUE
C      *** INITIALIZE RANGE VARIABLE AT WHICH SOLUTION IS TO BE PRINTED
       XPR=RA+PDR
C
C      *** INITIALIZE RANGE VARIABLE AT WHICH SOLUTION IS TO BE RECORDED
```

```
      XWR=RA+WDR
      IF(WDR.EQ.0.0) XWR=RA+RMAX+1.0
C
C     *** SAVE RANGE STEP
      OLDR=DR
C
C     *** INITIALIZE PARAMS FOR DIAG THEN COMPUTE MAIN DIAGONALS X AND Y
      N1=1
      DR1=DR
C     *** COMPUTE X AND Y DIAGONALS
C     *** ASSUMPTION IS THAT DENSITY AND SOUND SPEED ARE CONSTANT FROM
C     *** RA TO RA+DR.
      CALL DIAG
C
C     *** MAIN LOOP STARTS HERE ***
C     *** SOLUTION WILL BE ADVANCED FROM RANGE RA TO RANGE RA+DR
C
100   CONTINUE
      IDIAG=0
C
C     *** THIS SECTION OF CODE DETERMINES WHETHER OR NOT TO RESTORE STEP
C     *** SIZE TO THE ORIGINAL DR.
      IF(ITYPEB.EQ.3) GO TO 101
      IF(THETA.NE.0.0) GO TO 102
101   IF(DR.NE.OLDR) IDIAG=1
      DR=OLDR
102   CONTINUE
C
C     *** DOES SVP CHANGE BEFORE BOTTOM PROFILE?
      IF(RSVP.GE.R2) GO TO 105
C     *** YES. DOES SVP CHANGE BEFORE NEXT SOLUTION RANGE?
      IF(RA+DR.LE.RSVP) GO TO 131
C     *** YES. ADJUST DR SO THAT SOLUTION ADVANCES TO RSVP
      DR=RSVP-RA
      GO TO 126
C     *** BOTTOM PROFILE CHANGES BEFORE OR AT SAME RANGE AS SVP
105   CONTINUE
C
C     *** DETERMINE IF BOTTOM DEPTHS ARE TO BE UPDATED
C     *** FIRST PASS - RA WILL BE .LT. R2
C     *** UPDATE BOTTOM DEPTHS?
      IF(RA.GE.R2) GO TO 110
C
C     *** NO.
C     *** DETERMINE IF RANGE STEP TOO LARGE
      IF(RA+DR.LE.R2) GO TO 131
C     *** RANGE STEP IS TOO LARGE - RESET DR - ADV SOLUTION TO R2
      DR=R2-RA  ! ............
      GO TO 126
C
C     *** UPDATE BOTTOM DEPTHS
110   CONTINUE
      R1=R2
      Z1=Z2
      ITRK=ITRK+1
      R2=TRACK(ITRK,1)
      Z2=TRACK(ITRK,2)
C     *** TWO DEPTHS AT SAME RANGE INDICATE VERTICAL DISCONTINUITY.
C     *** ADVANCE TRACK FORWARD.
      IF(R1.EQ.R2) GO TO 110
```

```
C
C     *** RESTORE DR
      DR=OLDR
C
C     *** COMPUTE SLOPE OF BOTTOM
      THETA=ATAN2(Z2-Z1,R2-R1)
C
C     *** IF BOTTOM IS NOT FLAT, COMPUTE NEW RANGE STEP
      IF(THETA.EQ.0) GO TO 112
C
C     *** IF BOTTOM OF ARTIFICIAL LAYER IS FLAT, DO NOT RECOMPUTE DR.
      IF(ITYPEB.EQ.3) GO TO 112
      DR=ABS(DZ*COS(THETA)/SIN(THETA))
C
C     *** IF RANGE STEP TOO LARGE, PRINT WARNING MESSAGE. RECOMPUTE DR.
112   IF(RA+DR.LE.R2) GO TO 120
      WRITE(NPU,115)
115   FORMAT(1X,'RANGE STEP TOO LARGE FOR BOTTOM IRREGULARITIES')
C     STOP  !.....................................
      DR=R2-RA ! ...................
C
120   CONTINUE
      IF(RA+DR.GT.RSVP) DR=RSVP-RA
C
C     *** PRINT BOTTOM DEPTHS
      IF(IBOT.EQ.0) GO TO 126
      TH=THETA*DEG
      WRITE(NPU,123) R1,Z1,R2,Z2,TH
123   FORMAT(//1X,'BOTTOM DEPTHS ',//,1X,
     C'R1    = ',F10.1,' M',/,1X,
     C'Z1    = ',F8.1,' M',/,1X,
     C'R2    = ',F10.1,' M',/,1X,
     C'Z2    = ',F8.1,' M',/,1X,
     C'THETA = ',F8.1,' DEG')
126   CONTINUE
      DZZ=DR*TAN(THETA)
      WRITE(NPU,128) DR,RA
128   FORMAT(1X,'RANGE STEP = ',F8.2,' M AT RANGE ',F10.1,' M')
C
C     *** ADVANCE RANGE ONE STEP
130   RA=RA+DR
      IDIAG=1
      GO TO 132
131   CONTINUE
      RA=RA+DR
132   CONTINUE
C     *** READ NEW SVP PROFILE FLAG?
      IF(RA.GE.RSVP) READ(NIU,*,END=33) KSVP
C
C     *** IF KSVP NOT 0, SUBROUTINE USVP IS CALLED FOR SVP. USER IS
C     *** RESPONSIBLE FOR ADJUSTING DEPTHS OF LAYERS WHEN BOTTOM
C     *** IS NOT FLAT.
      IF(KSVP.EQ.0) GO TO 134
      CALL USVP
      IDIAG=1
      GO TO 148
C
C     *** NEW SVP AT ADVANCED RANGE?
134   IF(RA.GE.RSVP) GO TO 145  !.....................
C
```

```
C     *** NO. IS BOTTOM FLAT?
135   IF(THETA.EQ.0.0) GO TO 166
      MLYR=NLYR
      IF(ITYPEB.EQ.3) MLYR=NLYR-1
C
C     *** UPDATE DEPTHS OF LAYERS
C     *** ASSUMES LAYERS ARE PARALLEL AND FOLLOW BOTTOM CONTOUR
C     *** IF ITYPEB = 3, BOTTOM OF ARTIFICIAL LAYER REMAINS FLAT.
      DO 138 ILYR=1,MLYR
      ZLYR(ILYR)=ZLYR(ILYR)+DZZ
138   CONTINUE
C     N1=ZLYR(1)/DZ
      N1=1
      ZA=ZLYR(NLYR)
      IF(ITYPEB.EQ.3.AND.ZLYR(NLYR).LT.ZLYR(MLYR))WRITE(NPU,139)
139   FORMAT(1X,'ERR. DEPTH OF ARTIFICIAL LAYER LESS THAN LAYER ABOVE',
     C/,1X,'MAKE ARTIFICIAL LAYER DEEPER.')
C
C     *** ADJUST DEPTHS OF PROFILES IN SLOPING LAYERS
C     *** ASSUMES SAME SVP IN LAYERS
      NWSVP=IXSVP(1)
      DO 140 I=NWSVP+1,NSVP
      ZSVP(I)=ZSVP(I)+DZZ
140   CONTINUE
      GO TO 167
C
C     *** GET NEXT SVP
145   CONTINUE
      CALL SVP(NLYR,ZLYR,RHO,BETA,IXSVP,NSVP,ZSVP,CSVP)
      IDIAG=1
148   CONTINUE
C     *** ERROR DETECTED?
      IF(NSVP.EQ.0) GO TO 33
      IF(ITYPEB.NE.3) ZA=ZA+DZZ
C     *** NO. READ RANGE OF NEXT PROFILE?
      IF(RA.GE.RSVP) READ(NIU,*,END=150) RSVP
149   CONTINUE
C     *** ARTIFICIAL ABSORBING LAYER?
      IF(ITYPEB.LT.2) GO TO 155
      IF(ITYPEB.GT.3) GO TO 155
C     *** YES. UPDATE DENSITY, ATTEN AND SPEED.
      IF(ZA.LT.ZLYR(NLYR)) GO TO 155
      NLYR=NLYR+1
      ZLYR(NLYR)=ZA
      RHO(NLYR)=RHO(NLYR-1)
      BETA(NLYR)=BETA(NLYR-1)
      NSVP=NSVP+1
      IXSVP(NLYR)=NSVP
      ZSVP(NSVP)=ZLYR(NLYR)
      CSVP(NSVP)=CSVP(NSVP-1)
      GO TO 155
C     *** SET RSVP LARGE SO THAT LAST PROFILE IS USED FOR REMAINDER OF PROBLEM
150   RSVP=1.0E+38
      GO TO 149
C
C     *** PRINT SVP
155   IF(ISVP.EQ.0) GO TO 165
      WRITE(NPU,93) RA
      ILYR=1      !!!!!!!!!!!!!!!!!!!!
      DO 160 I=1,NSVP
```

```
      WRITE(NPU,94) I,ZSVP(I),CSVP(I)
C      IF(I.NE.IXSVP(ILYR)) GO TO 160 !!!!!!!!!!!!!!
C      WRITE(NPU,84)ILYR,ZLYR(ILYR),RHO(ILYR),BETA(ILYR)!!!!!!!!!!!!!!!!!!!
      ILYR=ILYR+1 !!!!!!!!!!!!!!!!!
160   CONTINUE
165   CONTINUE
C
C     *** IF NEW DR AND/OR SVP - UPDATE X AND Y DIAGONALS
166   IF(IDIAG.EQ.0) GO TO 170
      N1=1
167   CALL DIAG
      DR1=DR
170   CONTINUE
C
C     *** ADVANCE SOLUTION ONE STEP FORWARD
      NOLD=N
      CALL CRNK
C
C     *** APPLY ABSORPTION IF ITYPEB = 2 OR 3
      IF(ITYPEB.LT.2) GO TO 185
      IF(ITYPEB.GT.3) GO TO 185
      MM=ZLYR(NLYR-1)/DZ
      NA=N-MM
      IF(NA.GT.0) GO TO 175
      IF(NA.EQ.0) GO TO 185
      WRITE(NPU,174) RA
174   FORMAT(1X,'ERR IN THICKNESS OF ABSORBING LAYER AT ',F8.1,' M')
      STOP
175   CONTINUE
C     *** SEE AESD PE MODEL BY BROCK - NORDA TECH NOTE 12 - JAN 78
      DO 180 I=1,NA
      ATT=EXP(-.01*DR*EXP(-((I-NA)/(NA/3.0))**2.0))
      U(MM+I)=U(MM+I)*ATT
180   CONTINUE
185   CONTINUE
C
C     *** TERMINATE RUN IF N HAS BEEN DECREMENTED TO 5 - ARBITRARY
      IF(N.GT.5) GO TO 200
      WRITE(NPU,190)
190   FORMAT(1X,'PROGRAM TERMINATED - ONLY FIVE POINTS REMAIN')
      GO TO 400
200   CONTINUE
C
C     *** TERMINATE RUN IF N HAS BEEN INCREMENTED BEYOND MAXIMUM
      IF(N.LE.MXN) GO TO 250
      WRITE(NPU,210)RA
210   FORMAT(1X,'MAX N EXCEEDED AT RANGE ',F10.2,' M.')
      GO TO 400
250   CONTINUE
C
C     *** DETERMINE IF NEW POINT ADDED
      IF(N.LE.NOLD) GO TO 255
      N1=NOLD
      CALL DIAG
      DR1=DR
255   CONTINUE
C
C     *** IF SOLUTION IS TO BE WRITTEN ON DISK,
C     *** WRITE SELECTED PRESSURE FIELD AT RANGE RA
      IF(XWR.GT.RA) GO TO 260
```

```
C     *** RECOMPUTE NZ. VARIABLE DZ TO BE INCORPORATED AT LATER DATE.
      IWZ=WDZ/DZ+.5
      WDZ=IWZ*DZ
      NZ=N/IWZ
      WRITE(NOU) NZ,RA,WDZ,(U(I),I=IWZ,N,IWZ)
C
C     *** DETERMINE NEXT RANGE AT WHICH TO WRITE SOLUTION ON DISK
      XWR=XWR+WDR
260   CONTINUE
C
C     *** DETERMINE IF SOLUTION IS TO BE PRINTED
      IF(XPR.GT.RA.OR.IPZ.EQ.0.OR.PDR.EQ.0.0) GO TO 350
C
C     *** PRINT RANGE
      RTEMP=RA/1000.0
      WRITE(NPU,270) RTEMP
270   FORMAT(/5X,'RANGE =',E15.8,' KM. '/)
C
C     *** COMPUTE HANKEL FUNCTION
      IF(IHNK.NE.0) HNK=HNKL(XK0*RA)
C
C     *** COMPUTE AND PRINT PROPAGATION LOSS AT EACH IPZ'TH DEPTH
      WRITE(NPU,275)
275   FORMAT(6X,'I',9X,'Z(I)',6X,'LOSS(DB)',14X,'U(I)')
      DO 300 I=IPZ,N,IPZ
      ZI=I*DZ
      PL=CABS(U(I))
      IF(IHNK.EQ.1) PL=CABS(U(I)*HNK)
      IF(PL.LE.0.0) GO TO 288
      PL=-20.0*ALOG10(PL)
      IF(IHNK.EQ.0) PL=PL+10.0*ALOG10(RA)
      GO TO 289
288   PL=999.9
289   WRITE(NPU,295) I,ZI,PL,U(I)
295   FORMAT(2X,I5,(3X,F10.2,3X,F10.3),3X,'(',E12.5,2X,E12.5,' )')
300   CONTINUE
C
C     *** DETERMINE NEXT RANGE AT WHICH TO PRINT SOLUTION
      XPR=XPR+PDR
C
C     *** TERMINATE RUN IF SOLUTION AT MAXIMUM RANGE HAS BEEN OBTAINED
350   IF(RA.LT.RMAX) GO TO 100
C
C     *** TERMINATE RUN
400   IF(WDR.NE.0)CALL CLOSE(NOU)
      WRITE(NPU,10) TM
      CALL TIME(TM)
      WRITE(NPU,10) TM
      WRITE(NPU,401)
401   FORMAT(1X,'END OF RUN')
      STOP
      END
```

```
      SUBROUTINE BCON
C     ***********************************************************************
C     *** USER PREPARED BOTTOM CONDITION SUBROUTINE
C         BCON IS CALLED IF INPUT PARAMETER ITYPEB = 1
C         SEE MAIN PROGRAM FOR DEFINITIONS
C     ***********************************************************************
C     *** SUBROUTINE RETURNS:
C         BOTY,BOTX
C     ***********************************************************************
C
      PARAMETER MXLYR=101,MXN=10000,MXSVP=101,MXTRK=101,NIU=1,
     C          NOU=2,NPU=6
      COMPLEX ACOFX,ACOFY,BCOF,BOTX,BOTY,BTA,HNK,HNKL,SURX,SURY,TEMP,
     C        U,X,Y
      COMMON /IFDCOM/ACOFX,ACOFY,ALPHA,BCOF,BETA(MXLYR),BOTX,BOTY,
     C       BTA(MXN),C0,CSVP(MXSVP),DR,DR1,DZ,FRQ,IHNK,ISF,ITYPEB,
     C       ITYPES,IXSVP(MXLYR),KSVP,N,N1,NLYR,NSVP,NWSVP,R12(MXN),RA,
     C       RHO(MXLYR),RSVP,SURX,SURY,THETA,TRACK(MXTRK,2),U(MXN),
     C       X(MXN),XK0,Y(MXN),ZA,ZLYR(MXLYR),ZP,ZS,ZSVP(MXSVP)
      DATA PI/3.141592654/,DEG/57.29578/
C
      IF(THETA) 50,100,150
C
C     *** THETA LESS THAN 0.0. BOTTOM SLOPES UP.
50    CONTINUE
      BOTY=U(N)
C     BOTX=.............
      RETURN
C
C     *** THETA = 0.0. BOTTOM IS FLAT.
100   CONTINUE
      BOTY=U(N)
C     BOTX=.............
      RETURN
C
C     *** THETA GREATER THAN 0.0, BOTTOM SLOPES DOWN.
150   CONTINUE
C     BOTY=.............
C     BOTX=.............
      RETURN
      END
```

```
      SUBROUTINE CRNK
C     ***********************************************************************
C     ***********************************************************************
C     * THIS ROUTINE USES THE CRANK-NICOLSON METHOD FOR SOLVING THE     *
C     * SYSTEM OF EQUATIONS. THE SOLUTION IS ADVANCED FROM RANGE RA-DR *
C     * TO RANGE RA WHERE RA IS THE ADVANCED RANGE.                     *
C     ***********************************************************************
C     ***********************************************************************
C     *** SYSTEM OF EQUATIONS IS AS FOLLOWS:
C
C           ALL VALUES ARE AT ADVANCED RANGE
C     ***                      ***     ***  ***     ***          ***
C     * X(1)  +R12(1)    0     *     * U(1) *     *    SURX     *
C     *  +1    X(2)    +R12(2) *  .  * U(2) *  +  *     0       *  =
C     *   0     +1      X(N)   *     * U(N) *     * R12(N)*BOTX *
C     ***                      ***     ***  ***     ***          ***
C
C           ALL VALUES ARE AT PRESENT RANGE
C     ***                      ***     ***  ***     ***          ***
C     * Y(1)  W*R12(1)   0     *     * U(1) *     *    W*SURY     *
C     *  W     Y(2)   W*R12(2) *  .  * U(2) *  +  *      0       *
C     *  0     W       Y(N)   *     * U(N) *     * W*R12(N)*BOTY *
C     ***                      ***     ***  ***     ***          ***
C
C     ***********************************************************************
C     ***********************************************************************
C
C         D      - ARRAY - RIGHT HAND SIDE
C         RA     - RANGE TO WHICH SOLUTION IS TO BE ADVANCED - METERS
C                - RA IS INCREMENTED BY DR PRIOR TO ENTERING THIS ROUTINE
C         TA     - TEMPORARY VARIABLE USED IN MANIPULATION OF BOTTOM POINT
C         TB     - TEMPORARY VARIABLE USED IN MANIPULATION OF BOTTOM POINT
C
C     *** SEE MAIN PROGRAM FOR OTHER DEFINITIONS
C     *** UNDEFINED VARIABLES ARE TEMPORARY VARIABLES
C     *** SUBROUTINE CRNK RETURNS:
C         N      - NUMBER OF EQUI-SPACED POINTS IN U AT RANGE RA
C         U      - ARRAY - COMPLEX ACOUSTIC PRESSURE FIELD AT RANGE RA
C                - INCLUDES BOTTOM POINT - DOES NOT INCLUDE SURFACE POINT
C     *** SUBROUTINE TRID IS CALLED TO SOLVE THE TRIDIAGONAL MATRIX
C     ***********************************************************************
C
      PARAMETER MXLYR=101,MXN=10000,MXSVP=101,MXTRK=101,NIU=1,
     C          NOU=2,NPU=6
      COMPLEX ACOFX,ACOFY,BCOF,BOTX,BOTY,BTA,HNK,HNKL,SURX,SURY,TEMP,
     C        U,X,Y
      COMMON /IFDCOM/ACOFX,ACOFY,ALPHA,BCOF,BETA(MXLYR),BOTX,BOTY,
     C       BTA(MXN),C0,CSVP(MXSVP),DR,DR1,DZ,FRQ,IHNK,ISF,ITYPEB,
     C       ITYPES,IXSVP(MXLYR),KSVP,N,N1,NLYR,NSVP,NWSVP,R12(MXN),RA,
     C       RHO(MXLYR),RSVP,SURX,SURY,THETA,TRACK(MXTRK,2),U(MXN),
     C       X(MXN),XK0,Y(MXN),ZA,ZLYR(MXLYR),ZP,ZS,ZSVP(MXSVP)
      DATA PI/3.141592654/,DEG/57.29578/
      COMPLEX W,W1,W1S,W2,W2S,XN2,XNW,EYE,G,H
      COMMON /WIDANG/W,WA,WB,WC,WD,W1,W1S,W2,W2S,XN2
      DIMENSION D(MXN)
      COMPLEX D,SUM,A,EX,P,Q,XX,S1,S2,CA,CB,CC,CD,TA,TB
      DATA EYE/(0.0,1.0)/
C
C     *** CALL SCON FOR SURFACE CONDITION
      CALL SCON
```

```
C
C     *** COMPUTE RIGHT HAND SIDE D(1) THROUGH D(N-1)
C     *** D(N) IS COMPUTED LATER
C     *** W IS COMPUTED IN SUBROUTINE DIAG
      D(1)=Y(1)*U(1)+W*R12(1)*U(2)+W*SURY-SURX
      DO 5 I=2,N-1
5     D(I)=W*U(I-1)+Y(I)*U(I)+W*R12(I)*U(I+1)
C
C     *** BOTTOM TYPE
      IB=ITYPEB+1
      GO TO (10,40,80,82) ,IB
C
C     *** BOTTOM IS RIGID - ITYPEB = 0
10    CONTINUE
      IF(THETA.GT.0.0) GO TO 30
      IF(THETA.NE.0.0) GO TO 15
C
C     *** RIGID BOTTOM IS FLAT
      BOTY=U(N)
      D(N)=W*U(N-1)+Y(N)*U(N)+W*R12(N)*BOTY
      TA=0.0
C     *** BOTX=U(N) AT ADVANCED RANGE
      TB=1.0
      CALL TRID(X,U,D,BTA,R12,N,N,TA,TB)
      RETURN
C
C     *** RIGID BOTTOM SLOPES UPWARD - DELETE A POINT - USE 2ND ORDER ODE
15    BOTY=U(N)
      N=N-1
      D(N)=W*U(N-1)+Y(N)*U(N)+W*R12(N)*BOTY
      COT=COS(THETA)/SIN(THETA)
      XNW=XN2-1.0
      G=(EYE*XK0*(1.0-WA/WC+XNW*(-WB/WC+WA*WD/(WC*WC))+XNW*XNW*
     C(WB*WD/(WC*WC)-WA*WD*WD/(WC*WC*WC))+XNW*XNW*XNW*(-WB*WD*WD/
     C(WC*WC*WC))))
      H=EYE*XK0*(-WB/WC+WA*WD/(WC*WC)+XNW*(2.0*WB*WD/(WC*WC)+
     C2.0*WA*WD*WD/(WC*WC*WC))+XNW*XNW*(-3.0*WB*WD*WD/(WC*WC*WC)))
      Q=(EYE*XK0+G)/H
      P=-COT/H
      XX=CSQRT(P*P-4.0*Q)
      S1=(-P+XX)/2.0
      S2=(-P-XX)/2.0
      CA=(1.0-S1*DZ)/(-XX*DZ)
      CB=1.0/(XX*DZ)
      CC=1.0-CA
      CD=CB
      TA=-CD*CEXP(S1*DZ)+CB*CEXP(S2*DZ)
      TB=CC*CEXP(S1*DZ)+CA*CEXP(S2*DZ)
      CALL TRID(X,U,D,BTA,R12,N,N,TA,TB)
      RETURN
C
30    CONTINUE
C     *** RIGID BOTTOM SLOPES DOWNWARD - USE 2ND ORDER O.D.E.
      COT=COS(THETA)/SIN(THETA)
      XNW=XN2-1.0
      G=(EYE*XK0*(1.0-WA/WC+XNW*(-WB/WC+WA*WD/(WC*WC))+XNW*XNW*
     C(WB*WD/(WC*WC)-WA*WD*WD/(WC*WC*WC))+XNW*XNW*XNW*(-WB*WD*WD/
     C(WC*WC*WC))))
      H=EYE*XK0*(-WB/WC+WA*WD/(WC*WC)+XNW*(2.0*WB*WD/(WC*WC)+
     C2.0*WA*WD*WD/(WC*WC*WC))+XNW*XNW*(-3.0*WB*WD*WD/(WC*WC*WC)))
```

```
      Q=(EYE*XK0+G)/H
      P=-COT/H
      XX=CSQRT(P*P-4.0*Q)
      S1=(-P+XX)/2.0
      S2=(-P-XX)/2.0
      CA=(1.0-S1*DZ)/(-XX*DZ)
      CB=1.0/(XX*DZ)
      CC=1.0-CA
      CD=CB
      TA=-CD*CEXP(S1*DZ)+CB*CEXP(S2*DZ)
      TB=CC*CEXP(S1*DZ)+CA*CEXP(S2*DZ)
C
C     *** COMPUTE BOTY - THEN COMPUTE D(N)
      BOTY=TA*U(N-1)+TB*U(N)
      D(N)=W*U(N-1)+Y(N)*U(N)+W*R12(N)*BOTY
C
C     *** COMPUTE TA AND TB FOR LOWER AND X DIAGONALS IN ROW N.
C     *** TA AND TB ARE COEFFICIENTS OF U(N-1) AND U(N) . USED IN THE
C     *** APPROXIMATION OF BOTX.
      XX=CSQRT(P*P-4.0*Q)
      S1=(-P+XX)/2.0
      S2=(-P-XX)/2.0
      CA=(1.0-S1*DZ)/(-XX*DZ)
      CB=1.0/(XX*DZ)
      CC=1.0-CA
      CD=CB
      TA=-CD*CEXP(S1*DZ)+CB*CEXP(S2*DZ)
      TB=CC*CEXP(S1*DZ)+CA*CEXP(S2*DZ)
      CALL TRID(X,U,D,BTA,R12,N,N,TA,TB)
C
C     *** ADD A POINT
      CB=((U(N)-U(N-1))/DZ-S1*U(N))/(-XX)
      CA=U(N)-CB
      N=N+1
      U(N)=CA*CEXP(S1*DZ)+CB*CEXP(S2*DZ)
      RETURN
C
40    CONTINUE
C
C     *** BOTTOM CONDITION SUPPLIED BY USER - ITYPEB = 1
      IF(THETA.GT.0.0) GO TO 60
      IF(THETA.NE.0.0) GO TO 70
C
C     *** FLAT BOTTOM
C     *** USER SUPPLIES BOTY ON BOTTOM INTERFACE AT PRESENT RANGE
C     *** USER SUPPLIES BOTX ON BOTTOM INTERFACE AT ADVANCED RANGE
50    CALL BCON
      TA=0.0
      TB=0.0
      N=N-1
      D(N)=W*U(N-1)+Y(N)*U(N)+W*R12(N)*BOTY-R12(N)*BOTX
      CALL TRID(X,U,D,BTA,R12,N,N,TA,TB)
      N=N+1
      U(N)=BOTX
      RETURN
C
60    CONTINUE
C     *** LAYER SLOPES DOWN
C     *** USER SUPPLIES :
C         BOTY - ONE DZ IN BOTTOM AT PRESENT RANGE
```

```
C          BOTX - ON BOTTOM INTERFACE AT ADVANCED RANGE
      CALL BCON
      TA=0.0
      TB=0.0
      D(N)=W*U(N-1)+Y(N)*U(N)+W*R12(N)*BOTY-R12(N)*BOTX
      CALL TRID(X,U,D,BTA,R12,N,N,TA,TB)
      N=N+1
      U(N)=BOTX
      RETURN
C
70    CONTINUE
C     *** LAYER SLOPES UP
C     *** USER SUPPLIES :
C          BOTX - ONE DZ BELOW BOTTOM INTERFACE AT ADVANCED RANGE
      CALL BCON
      N=N-1
      D(N)=W*U(N-1)+Y(N)*U(N)+W*R12(N)*BOTY-R12(N)*BOTX
      TA=0.0
      TB=0.0
      CALL TRID(X,U,D,BTA,R12,N,N,TA,TB)
      RETURN
C
C
C     *** ITYPEB = 2 -- ARTIFICIAL ABSORBING LAYER INTRODUCED
80    CONTINUE
      IF(THETA.NE.0.0) GO TO 85
C
C     *** BOTTOM OF ABSORBING LAYER IS FLAT
C     *** ITYPEB = 2 OR 3 -- ARTIFICIAL ABSORBING LAYER
82    U(N)=0.0
      TA=0.0
      TB=0.0
      CALL TRID(X,U,D,BTA,R12,N-1,N,TA,TB)
      RETURN
C
C     *** ITYPEB = 2  -- ARTIFICIAL ABSORBING LAYER NOT FLAT
85    IF(THETA.GT.0.0) GO TO 90
C
C     *** BOTTOM OF ABSORBING LAYER  SLOPES UP
      TA=0.0
      TB=0.0
      U(N-1)=0.0
      U(N)=0.0
      CALL TRID(X,U,D,BTA,R12,N-2,N-1,TA,TB)
C
C     *** DELETE A POINT
      N=N-1
      RETURN
C
90    CONTINUE
C     *** BOTTOM OF ABSORBING LAYER SLOPES DOWN
      D(N)=W*U(N-1)
      TA=0.0
      TB=0.0
      CALL TRID(X,U,D,BTA,R12,N,N,TA,TB)
C     *** ADD A POINT
      N=N+1
      U(N)=0.0
      RETURN
      END
```

```
      SUBROUTINE DIAG
C     ********************************************************************
C     ********************************************************************
C     * COMPUTES RANGE AND DEPTH DEPENDENT MAIN DIAGONALS OF MATRICES.  *
C     * HOWEVER, ASSUMPTION IS THAT DENSITY AND SOUND SPEED ARE         *
C     * CONSTANT FROM RA TO RA+DR.                                      *
C     * DIAG IS CALLED WHENEVER BOTTOM DEPTH OR SVP CHANGES             *
C     ********************************************************************
C     ********************************************************************
C
C     *** SEE MAIN PROGRAM FOR DEFINITIONS
C     *** SUBROUTINE DIAG RETURNS:
C         ACOFX - COEFFICIENT 'A' AT BOTTOM - AT RANGE RA
C         ACOFY - COEFFICIENT 'A' AT BOTTOM - AT RANGE RA-DR
C         BTA   - ARRAY - PARTIAL SOLUTION OF SYSTEM OF EQUATIONS
C         X     - ARRAY - MAIN DIAGONAL OF MATRIX AT RANGE RA
C         Y     - ARRAY - MAIN DIAGONAL OF MATRIX AT RANGE RA-DR
C     ********************************************************************
C
      PARAMETER MXLYR=101,MXN=10000,MXSVP=101,MXTRK=101,NIU=1,
     C          NOU=2,NPU=6
      COMPLEX ACOFX,ACOFY,BCOF,BOTX,BOTY,BTA,HNK,HNKL,SURX,SURY,TEMP,
     C        U,X,Y
      COMMON /IFDCOM/ACOFX,ACOFY,ALPHA,BCOF,BETA(MXLYR),BOTX,BOTY,
     C       BTA(MXN),C0,CSVP(MXSVP),DR,DR1,DZ,FRQ,IHNK,ISF,ITYPEB,
     C       ITYPES,IXSVP(MXLYR),KSVP,N,N1,NLYR,NSVP,NWSVP,R12(MXN),RA,
     C       RHO(MXLYR),RSVP,SURX,SURY,THETA,TRACK(MXTRK,2),U(MXN),
     C       X(MXN),XK0,Y(MXN),ZA,ZLYR(MXLYR),ZP,ZS,ZSVP(MXSVP)
C
      COMPLEX W,W1,W1S,W2,W2S
      COMMON /WIDANG/W,WA,WB,WC,WD,W1,W1S,W2,W2S,XN2
      DATA PI/3.141592654/,DEG/57.29578/
      COMPLEX B,XN1,XN2,EYE
      DATA IEX/0/
C
C     *** COMPUTE NEW X AND Y DIAGONALS
C
      EYE=CMPLX(0.0,1.0)
      W1=CMPLX(WC,+.5*XK0*DR*(WA-WC))
      W1S=CMPLX(WC,-.5*XK0*DR*(WA-WC))
      W2=CMPLX(WD,+.5*XK0*DR*(WB-WD))
      W2S=CMPLX(WD,-.5*XK0*DR*(WB-WD))
      W=W2/W2S
      DZK=.5*DZ*DZ*XK0*XK0
C     *** GET INDICES, DENSITY, AND ATTENUATION FOR FIRST LAYER
      ILYR=1
      L=1
      M=IXSVP(1)
      R1=RHO(1)
      BETA1=BETA(1)
      DO 100 I=N1,N
C     *** ZI IS RECEIVER DEPTH
      ZI=I*DZ
C     *** IS RECEIVER IN THIS LAYER?
20    IF(ZI.LE.ZLYR(ILYR)) GO TO 49
C     *** NO. ALL LAYERS CHECKED?
      IF(ILYR.EQ.NLYR) GO TO 52
C     *** NO. SET INDICES, DENSITY, AND ATTENUATION FOR NEXT LAYER.
      ILYR=ILYR+1
      L=M+1
```

```
      M=IXSVP(ILYR)
      R1=RHO(ILYR)
      BETA1=BETA(ILYR)
      GO TO 20
C     *** DEPTH ZI IS IN THIS LAYER.
49    CONTINUE
C     *** DETERMINE SOUND SPEED CI AT DEPTH ZI
      DO 50 J=L,M
      IF(ZI.GT.ZSVP(J)) GO TO 50
      L=J
C     *** INTERPOLATE
      CI=CSVP(J-1)+(CSVP(J)-CSVP(J-1))*(ZI-ZSVP(J-1))/(ZSVP(J)-ZSVP(J-1)
     C)
      GO TO 60
50    CONTINUE
C     *** EXTRAPOLATE
52    CONTINUE
      IF(ZSVP(M).NE.ZSVP(M-1))GO TO 53
      CI=CSVP(M)
      GO TO 54
53    CI=CSVP(M-1)+(CSVP(M)-CSVP(M-1))*(ZI-ZSVP(M-1))/
     C(ZSVP(M)-ZSVP(M-1))
54    IF(IEX.EQ.1) GO TO 60
      WRITE(NPU,56)
56    FORMAT(1X,'WARNING. EXTRAPOLATION OF SVP PERFORMED.')
      IEX=1
60    CONTINUE
C     *** SAVE SPEED IN MEDIUM 1
      C1=CI
C     *** IS RECEIVER ON OR WITHIN 1 DZ OF INTERFACE?
      IF(ZLYR(ILYR)-ZI.GE.DZ) GO TO 65
C     *** YES. IS THIS BOTTOM LAYER?
      IF(ILYR.EQ.NLYR) GO TO 65
C     *** NO. GET PARAMETERS FOR MEDIUM 2
      R2=RHO(ILYR+1)
      BETA2=BETA(ILYR+1)
      C2=CSVP(IXSVP(ILYR)+1)
      GO TO 70
65    CONTINUE
C     *** NOT AN INTERFACE. USE MEDIUM 1 PARAMETERS.
      C2=C1
      R2=R1
      BETA2=BETA1
C     *** COMPUTE DENSITY RATIO
70    R12(I)=R1/R2
C     *** THE NEXT FEW LINES COMPUTE THE X DIAGONAL
      XN=C0/C1
      IF(BETA1.LT.0.0) BETA1=ALPHA*C1/FRQ
      XN1=CMPLX(XN*XN,XN*XN*BETA1/27.287527)
      XN=C0/C2
      IF(BETA2.LT.0.0) BETA2=ALPHA*C2/FRQ
      XN2=CMPLX(XN*XN,XN*XN*BETA2/27.287527)
      B=(DZK*((XN1-1.)+R12(I)*(XN2-1.0)))
      R=(1.0+R12(I))
      X(I)=W1S/W2S*DZK*R-R+B
      Y(I)=W1/W2S*DZK*R-W*R+W*B
100   CONTINUE
C
C     *** COMPUTE BTA(N1) THROUGH BTA(N)
C     *** ARRAY BTA CONTAINS PARTIAL SOLUTION OF SYSTEM OF EQUATIONS
```

```
      IF(N1.EQ.1) GO TO 105
      M=N1
      GO TO 106
105   BTA(1)=X(1)
      M=2
106   DO 110 I=M,N
      BTA(I)=X(I)-R12(I-1)/BTA(I-1)
110   CONTINUE
C
      RETURN
      END
```

```
      COMPLEX FUNCTION HNKL(X)
C     ************************************************************************
C     * HANKEL FUNCTION H0(1) - POLYNOMIAL APPROXIMATION                     *
C     * HANDBOOK OF MATH FUNCTIONS - N.B.S. - NOV 1967                       *
C     ************************************************************************
C
      REAL J0
      DATA PI/3.141592654/
C
      IF(X.GT.3.) GO TO 10
C
C     *** (-3.0.LE.X.LE.3.0)
      Y=X*X/9.0
      J0=1.+Y*(-2.2499997+Y*(+1.2656208+Y*(-0.3163866+Y*(+0.0444479
     C     +Y*(-0.0039444+Y*(+0.0002100))))))
C
C     *** (0.0.LT.X.LE.3.0)
      Y0=2.0*LOG(0.5*X)*J0/PI+0.36746691
     C     +Y*(+0.60559366+Y*(-0.74350384+Y*(+0.25300117+Y*(-0.04261214
     C     +Y*(+0.00427916+Y*(-0.00024846))))))
      HNKL=CMPLX(J0,Y0)
      RETURN
C
C     *** (3.0.LE.X.LT.INFINITY)
10    Y=3.0/X
      F0=  0.79788456+Y*(-0.00000077+Y*(-0.00552740+Y*(-0.00009512
     C               +Y*(+0.00137237+Y*(-0.00072805+Y*(+0.00014476))))))
      T0=X-0.78539816+Y*(-0.04166397+Y*(-0.00003954+Y*(+0.00262573
     C               +Y*(-0.00054125+Y*(-0.00029333+Y*(+0.00013558))))))
      HNKL=F0*CEXP(CMPLX(0.0,T0))/SQRT(X)
      RETURN
      END
```

```
      SUBROUTINE SCON
C     ******************************************************************
C     *** SURFACE CONDITION SUBROUTINE
C         IF ITYPES = 0, SCON SETS SURY AND SURX = 0.0.
C         IF ITYPES NOT 0, THE USER MUST SUPPLY SURY AND SURX.
C         SEE MAIN PROGRAM FOR DEFINITIONS
C     ******************************************************************
C
      PARAMETER MXLYR=101,MXN=10000,MXSVP=101,MXTRK=101,NIU=1,
     C          NOU=2,NPU=6
      COMPLEX ACOFX,ACOFY,BCOF,BOTX,BOTY,BTA,HNK,HNKL,SURX,SURY,TEMP,
     C        U,X,Y
      COMMON /IFDCOM/ACOFX,ACOFY,ALPHA,BCOF,BETA(MXLYR),BOTX,BOTY,
     C       BTA(MXN),C0,CSVP(MXSVP),DR,DR1,DZ,FRQ,IHNK,ISF,ITYPEB,
     C       ITYPES,IXSVP(MXLYR),KSVP,N,N1,NLYR,NSVP,NWSVP,R12(MXN),RA,
     C       RHO(MXLYR),RSVP,SURX,SURY,THETA,TRACK(MXTRK,2),U(MXN),
     C       X(MXN),XK0,Y(MXN),ZA,ZLYR(MXLYR),ZP,ZS,ZSVP(MXSVP)
      DATA PI/3.141592654/,DEG/57.29578/
C
      IF(ITYPES.NE.0) GO TO 100
C
C     *** PRESSURE RELEASE SURFACE
      SURY=0.0
      SURX=0.0
      RETURN
C
C     *** USER SURFACE CONDITION
100   CONTINUE
C     SURY=.........
C     SURX=.........
      RETURN
      END
```

```
      SUBROUTINE SFIELD(FRQ,C0,ZS,N,DZ,U)
C     ********************************************************************
C     *** GAUSSIAN STARTING FIELD - SEE NORDA TECH NOTE 12 BY H.K.BROCK
C     *** SFIELD IS CALLED IF INPUT PARAMETER ISF = 0
C     ********************************************************************
C
C     *** CALLING ROUTINE SUPPLIES:
C         FRQ - FREQUENCY IN HZ
C         C0  - REFERENCE SOUND SPEED - METERS/SEC
C         ZS  - DEPTH OF SOURCE IN METERS.
C         N   - NUMBER OF POINTS IN ARRAY U
C         DZ  - DEPTH INCREMENT - METERS
C     *** SFIELD SUBROUTINE SUPPLIES:
C         U   - COMPLEX STARTING FIELD
C     ********************************************************************
C
      COMPLEX U(1)
      DATA PI/3.1415926535/
C
C     THE FIELD IS DEFINED AS A GAUSSIAN BEAM AT RANGE = 0.
C     LOCAL VARIABLES - GA  GAUSSIAN AMPLITUDE
      XK0=2.0*PI*FRQ/C0
      GW=2.0/XK0
      GA=SQRT(GW)/GW
      DO 10 I=1,N
      ZM=I*DZ
      PR=GAUSS(GA,ZM,ZS,GW)-GAUSS(GA,-ZM,ZS,GW)
      U(I)=CMPLX(PR,0.0)
10    CONTINUE
      RETURN
      END
      FUNCTION GAUSS(GA,Z,GD,GW)
C     INPUT  - GA  GAUSSIAN AMPLITUDE
C     OUTPUT - GAUSS = GA * EXP(-((Z - GD) / GW)**2)
C     TEMPORARY VARIABLE - V
      V=(Z-GD)/GW
      V=-(V*V)
      GAUSS=GA*EXP(V)
      RETURN
      END
```

```
      SUBROUTINE SVP(NLYR,ZLYR,RHO,BETA,IXSVP,NSVP,ZSVP,CSVP)
C     ******************************************************************
C     *** SOUND VELOCITY PROFILE SUBROUTINE
C     *** CALLING PROGRAM SUPPLIES: NOTHING
C     *** SVP SUBROUTINE RETURNS:
C         NLYR - NUMBER OF LAYERS. LAYER 1 IS WATER. OTHERS ARE SEDIMENT
C         ZLYR - ARRAY - DEPTH OF EACH LAYER. FIRST IS DEPTH OF WATER.
C         RHO  - ARRAY - DENSITY OF EACH LAYER. GRAMS/CUBIC CM
C         BETA - ARRAY - ATTENUATION IN EACH LAYER. DB/WAVELENGTH
C         IXSVP- ARRAY - CONTAINS POINTERS. POINTS TO LAST VALUE OF SVP
C                IN CORRESPONDING LAYER. SVP IS STORED IN ARRAYS ZSVP
C                AND CSVP. IXSVP(1) POINTS TO LAST SVP POINT IN WATER.
C                IXSVP(NLYR) POINTS TO LAST SVP POINT IN BOTTOM-MOST LAYER.
C         NSVP - NUMBER OF POINTS IN ZSVP AND CSVP. ZSVP AND CSVP
C                CONTAIN THE PROFILES FOR ALL LAYERS.
C         ZSVP - ARRAY - SVP DEPTHS - METERS
C         CSVP - ARRAY - SOUND SPEED - METERS/SEC
C     ******************************************************************
C
      PARAMETER NIU=1,NPU=6
      DIMENSION ZLYR(1),RHO(1),BETA(1),IXSVP(1),ZSVP(1),CSVP(1)
C
      NSVP=0
C     *** READ NUMBER OF LAYERS
      READ(NIU,*,END=100) NLYR
C     *** FIRST LAYER IS WATER. OTHERS ARE SEDIMENT.
      DO 55 I=1,NLYR
C     *** READ DEPTH OF LAYER, DENSITY AND ATTENUATION.
      READ(NIU,*,END=100) ZLYR(I),RHO(I),BETA(I)
C     *** READ PROFILE.
50    READ(NIU,*,END=100) ZV,CV
      NSVP=NSVP+1
      ZSVP(NSVP)=ZV
      CSVP(NSVP)=CV
      IF(ZV.LT.ZLYR(I)) GO TO 50
      IF(ZV.GT.ZLYR(I)) GO TO 100
      IXSVP(I)=NSVP
55    CONTINUE
      RETURN
C
C     *** ERROR EXIT
100   NSVP=0
      RETURN
      END
```

```
      SUBROUTINE TRID(X,U,D,BTA,R12,L,M,TA,TB)
C     ****************************************************************
C     *** SPECIALIZED VERSION OF METHOD PRESENTED ON PG 442 OF CARNAHAN
C     *** ET AL FOR SOLVING A TRIDIAGONAL MATRIX.
C     *** TRID RETURNS THE SOLUTION FIELD IN U
C     *** BTA IS PARTIAL SOLUTION COMPUTED IN ROUTINE DIAG
C     *** D CONTAINS R.H.S.
C     *** GAMMA - TEMPORARY STORAGE
C     *** TA    - TEMPORARY VARIABLE USED IN MANIPULATION OF BOTTOM POINT
C     *** TB    - TEMPORARY VARIABLE USED IN MANIPULATION OF BOTTOM POINT
C     ****************************************************************
C
      PARAMETER MXN=10000
      DIMENSION X(1),U(1),GAMMA(MXN),D(1),BTA(1),R12(1)
      COMPLEX X,U,GAMMA,D,BTA,TA,TB
C     *** X AND LOWER DIAGONAL IN ROW L MUST BE MODIFIED BY TB AND TA
      BTA(L)=(X(L)+TB*R12(L))-((1.0+TA*R12(L))*R12(L-1))/BTA(L-1)
      GAMMA(1)=D(1)/BTA(1)
      DO 10 I=2,L
      GAMMA(I)=(D(I)-GAMMA(I-1))/BTA(I)
10    CONTINUE
      GAMMA(L)=GAMMA(L)-(TA*R12(L)*GAMMA(L-1))/BTA(L)
C     *** IF L IS LESS THAN M, U(M) IS KNOWN.
      IF(L.EQ.M) U(M)=GAMMA(M)
      DO 70 I=M,2,-1
      U(I-1)=GAMMA(I-1)-R12(I-1)*U(I)/BTA(I-1)
70    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE UFIELD
C     ****************************************************************
C     *** USER STARTING FIELD
C     *** USER WRITES THIS SUBROUTINE IF GAUSSIAN FIELD NOT DESIRED
C     *** UFIELD IS CALLED IF INPUT PARAMETER ISF IS NOT ZERO
C     ****************************************************************
C     *** UFIELD SUBROUTINE SUPPLIES:
C         U   - COMPLEX STARTING FIELD
C     ****************************************************************
      PARAMETER MXLYR=101,MXN=10000,MXSVP=101,MXTRK=101,NIU=1,
     C          NOU=2,NPU=6
      COMPLEX ACOFX,ACOFY,BCOF,BOTX,BOTY,BTA,HNK,HNKL,SURX,SURY,TEMP,
     C        U,X,Y
      COMMON /IFDCOM/ACOFX,ACOFY,ALPHA,BCOF,BETA(MXLYR),BOTX,BOTY,
     C       BTA(MXN),C0,CSVP(MXSVP),DR,DR1,DZ,FRQ,IHNK,ISF,ITYPEB,
     C       ITYPES,IXSVP(MXLYR),KSVP,N,N1,NLYR,NSVP,NWSVP,R12(MXN),RA,
     C       RHO(MXLYR),RSVP,SURX,SURY,THETA,TRACK(MXTRK,2),U(MXN),
     C       X(MXN),XK0,Y(MXN),ZA,ZLYR(MXLYR),ZP,ZS,ZSVP(MXSVP)
      DATA PI/3.141592654/,DEG/57.29578/
C
C     *** STARTING FIELD GENERATED BY USER
      DO 10 I=1,N
      ZI=I*DZ
C     U(I)=.....
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE USVP
C     ***********************************************************************
C     *** USER SOUND VELOCITY PROFILE SUBROUTINE
C         SUBROUTINE USVP IS CALLED EACH DR IN RANGE AS LONG AS
C         KSVP IS NOT ZERO. KSVP MAY BE USED BY USER TO TRANSFER CONTROL
C         IN THIS SUBROUTINE. USER INSERTS LOGIC TO CLEAR KSVP
C         WHEN USVP IS NO LONGER NEEDED. IF KSVP NOT CLEARED BY USER,
C         USVP IS CALLED EACH STEP IN RANGE UNTIL RA = NEXT RSVP.
C     ***********************************************************************
C     *** USVP SUBROUTINE RETURNS:
C         NLYR - NUMBER OF LAYERS. LAYER 1 IS WATER. OTHERS ARE SEDIMENT
C         ZLYR - ARRAY - DEPTH OF EACH LAYER. FIRST IS DEPTH OF WATER.
C         RHO  - ARRAY - DENSITY OF EACH LAYER. GRAMS/CUBIC CM
C         BETA - ARRAY - ATTENUATION IN EACH LAYER. DB/WAVELENGTH
C         IXSVP - ARRAY - CONTAINS POINTERS. POINTS TO LAST VALUE OF SVP
C                 IN CORRESPONDING LAYER. SVP IS STORED IN ARRAYS ZSVP
C                 AND CSVP. IXSVP(1) POINTS TO LAST SVP POINT IN WATER.
C         NSVP - NUMBER OF POINTS IN ZSVP AND CSVP. ZSVP AND CSVP
C                 CONTAIN THE PROFILES FOR ALL LAYERS.
C         ZSVP - ARRAY - SVP DEPTHS - METERS
C         CSVP - ARRAY - SOUND SPEED - METERS/SEC
C         KSVP - AS DESCRIBED ABOVE.
C     ***********************************************************************
C
      PARAMETER MXLYR=101,MXN=10000,MXSVP=101,MXTRK=101,NIU=1,
     C          NOU=2,NPU=6
      COMPLEX ACOFX,ACOFY,BCOF,BOTX,BOTY,BTA,HNK,HNKL,SURX,SURY,TEMP,
     C        U,X,Y
      COMMON /IFDCOM/ACOFX,ACOFY,ALPHA,BCOF,BETA(MXLYR),BOTX,BOTY,
     C       BTA(MXN),C0,CSVP(MXSVP),DR,DR1,DZ,FRQ,IHNK,ISF,ITYPEB,
     C       ITYPES,IXSVP(MXLYR),KSVP,N,N1,NLYR,NSVP,NWSVP,R12(MXN),RA,
     C       RHO(MXLYR),RSVP,SURX,SURY,THETA,TRACK(MXTRK,2),U(MXN),
     C       X(MXN),XK0,Y(MXN),ZA,ZLYR(MXLYR),ZP,ZS,ZSVP(MXSVP)
      DATA PI/3.141592654/,DEG/57.29578/
C
      GO TO (100,200,300,400) ,KSVP
      NSVP=0
      RETURN
C
100   CONTINUE
C
C     *** IF KSVP=1, CONTROL IS TRANSFERRED HERE. USER LOADS
C         NLYR,ZLYR(I),RHO(I),BETA(I), AND IXSVP(I) WHERE I=1,NLYR.
C         USER ALSO LOADS NSVP,ZSVP(I), AND CSVP(I) WHERE I=1,NSVP.
C         KSVP MAY BE ALTERED DEPENDING ON USER LOGIC.
C
C     *** USER SUPPLIES SVP
C     *** SETUP FOR ONE LAYER WITH FOUR SVP POINTS SHOWN BELOW
C
      NLYR=1
C     ZLYR(1)=.....
      RHO(1)=1.0
C     BETA(1)=.....
      NSVP=4
C     ZSVP(1)=.....
C     CSVP(1)=.....
C     ZSVP(2)=.....
C     CSVP(2)=.....
C     ZSVP(3)=.....
C     CSVP(3)=.....
```

```
C     ZSVP(4)=.....
C     CSVP(4)=.....
      IXSVP(1)=NSVP
      RETURN
C
200   CONTINUE
C     *** USER INSERTS CODE HERE IF DESIRED
      RETURN
C
300   CONTINUE
C     *** USER INSERTS CODE HERE IF DESIRED
      RETURN
C
400   CONTINUE
C     *** USER INSERTS CODE HERE IF DESIRED
      RETURN
      END
```

```
C     ************************************************************************
C     ************************************************************************
C     *** PLOT PROGRAM FOR IFD MODEL.
C     ************************************************************************
C     ************************************************************************
C     *  G. BOTSEAS, CODE 3332                                               *
C     *  NAVAL UNDERWATER SYSTEMS CENTER                                     *
C     *  NEW LONDON, CONNECTICUT 06320, U.S.A.                               *
C     ************************************************************************
C     ************************************************************************
C     *** PROGRAM PLOTS PROPAGATION LOSS VS RANGE ON CALCOMP PLOTTER -
C     *** MODEL 1039.
C     *** SEE MAIN PROGRAM IFD FOR DEFINITIONS OF IFD VARIABLES.
C     ************************************************************************
C     ************************************************************************
C     *** INPUT
C     ************************************************************************
C         INPUT UNIT NUMBER = NIU
C         INPUT FILE NAME   = PLTIFD.IN
C         CONTENTS:  CARD IMAGES IN FREE FORMAT
C         CARD 1   :  Z1,Z2,Z3,Y1,Y2,Y3,YL,X1,X2,X3,XL,FACT,XAVG
C           .      :   .          .          .          .          .
C         CARD N   :   .          .          .          .          .
C     ************************************************************************
C     *** QUICK REFERENCE AND NOTES FOR CARD INPUT
C     ************************************************************************
C         Z1 = FIRST RECEIVER DEPTH TO PLOT - METERS
C              IF (Z1.LE.0.0) PROGRAM TERMINATES
C         Z2 = LAST RECEIVER DEPTH TO PLOT  - METERS
C         Z3 = RECEIVER DEPTH INCREMENT - METERS
C         Y1 = LABEL OF Y-AXIS AT ORIGIN IN DB
C         Y2 = LABEL AT TOP OF Y-AXIS IN DB
C         Y3 = INCREMENT OF Y-AXIS LABELS IN DB
C         YL = LENGTH OF Y-AXIS IN INCHES
C         X1 = LABEL OF X-AXIS AT ORIGIN IN KILOMETERS
C         X2 = LABEL AT RIGHT OF X-AXIS IN KILOMETERS
C         X3 = INCREMENT OF X-AXIS LABELS IN KILOMETERS
C         XL = LENGTH OF X-AXIS IN INCHES
C         FACT = SCALE OF PLOT: 1.0 = FULL SIZE ; .5 = 1/2 SIZE ; ETC.
C         XAVG = RANGE OVER WHICH TO COMPUTE RUNNING AVERAGE IN METERS
C                IF XAVG = 0, ALL POINTS ARE PLOTTED
C     ************************************************************************
C     ************************************************************************
      PARAMETER MAXP=20000
      PARAMETER MXLYR=101,MXN=20000,NIU=1,NOU=2,NPU=6,PLTU=3
      COMPLEX HNK,HNKL,CTEMP,U(MXN)
      DIMENSION BETA(MXLYR),RHO(MXLYR),ZLYR(MXLYR),IBUF(2000)
      DIMENSION P(MAXP),R(MAXP)
      DATA PI/3.141592654/,DEG/57.29578/
      DATA CNVKM/1000.0/
      IPRNT=0
C     *** ASSIGN IFD OUTPUT FILE
      CALL ASSIGN(NOU,'IFD.OUT')
C     *** ASSIGN PLOT PARAMETER INPUT FILE
      CALL ASSIGN(NIU,'PLTIFD.IN')
C      CALL PLOTS(IBUF,2000,PLTU)
C       CALL PLOTS(0,0,1) !LNO3
C      CALL PLOT (0.0,0.5,-3)
C     *** READ PLOT PARAMETERS
100   READ(NIU,*,END=510) Z1,Z2,Z3,Y1,Y2,Y3,YL,X1,X2,X3,XL,FACT,XAVG
```

```
      IF(Z1.LE.0.0) GO TO 510
      IF(FACT.LE.0.0) FACT=1.0
C      CALL FACTOR(FACT)
      YINC=(Y2-Y1)/YL
      IF(Z3.EQ.0.0) Z3=1.0
       CALL PLOTS(0,0,1) !LNO3
      CALL PLOT (0.0,0.5,-3)
      CALL FACTOR(FACT)
C     *** GENERATE PLOT FOR RECEIVER DEPTH
      DO 350 ZR=Z1,Z2,Z3
      ZRR=ZR
      IPEN=3
      IX=1
      DX=(X2-X1)/XL
      CALL AXIS2(0.,0.,'RANGE (KM)',-10,XL,0.,X1,X3,X2)
      CALL AXIS2(XL,0.,' ',-1,YL,90.,Y1,Y3,Y2)
      CALL AXIS2(0.,0.,'PROPLOSS (DB)',+13,YL,90.,Y1,Y3,Y2)
      CALL PLOT(0.0,YL,3)
      CALL PLOT(XL,YL,2)
C     *** READ INITIAL IFD PARAMETERS
      REWIND(NOU)
      READ(NOU) FRQ,ZS,C0,ISF,R0,Z0,N,IHNK,ITYPEB,ITYPES,RMAX,DR,WDR,DZ,
     CNLYR,ZLYR,RHO,BETA
      IF(XAVG.LT.WDR) XAVG=WDR
      L=0
115   CONTINUE
      RAVG=0.0
      PLAVG=0.0
C     *** READ SOLUTION FIELD
120   READ(NOU,END=170)NN,RA,WDZ,(U(I),I=1,NN)
      IF(RA.LE.0.0) GO TO 120
      HNK=HNKL(2.0*PI*FRQ*RA/C0)
      I=0
      INTERP=0
      I=ZRR/WDZ
      IF(I.GT.NN) GO TO 115
      IF(I.GE.1) GO TO 130
      I=1
      ZRR=WDZ
130   IF(I*WDZ.NE.ZRR) INTERP=1
      IF(I.EQ.NN.AND.INTERP.EQ.1) GO TO 115
      Y=CABS(U(I))
      IF(IHNK.NE.0) Y=CABS(U(I)*HNK)
      IF(INTERP.EQ.1) CTEMP=U(I)+(U(I+1)-U(I))*(ZRR-I*WDZ)/WDZ
      IF(INTERP.EQ.1.AND.IHNK.NE.0) Y=CABS(CTEMP*HNK)
      IF(Y.LE.0.0) GO TO 120
      L=L+1
      P(L)=Y
      R(L)=RA
      GO TO 120
170   CONTINUE
      K=0
200   K=K+1
      RAVG=0
      PLAVG=0
      NAVG=0
      DO 210 J=K,L
      IF(R(J)-R(K).GE.XAVG) GO TO 220
      RAVG=RAVG+R(J)
      PLAVG=PLAVG+P(J)
```

```
      NAVG=NAVG+1
210   CONTINUE
220   CONTINUE
      IF(NAVG.EQ.0) GO TO 250
      BIAS=0.0
      RA=RAVG/NAVG
      IF(IHNK.EQ.0.AND.RA.GT.0.0) BIAS=10.0*ALOG10(RA)
      Y=PLAVG/NAVG
      IF(Y.LE.0.0) GO TO 200
      Y=-20.0*ALOG10(Y)+BIAS
      TEMP=RA/1000.0
      IF(IPRNT.EQ.1) WRITE(NPU,*) TEMP,Y
      IF(RA/CNVKM.GT.X2-XAVG/CNVKM) GO TO 250
      IF(RA/CNVKM.LT.X1) GO TO 200
      X=(RA/CNVKM-X1)/DX
      Y=(Y-Y1)/YINC
      IF(Y.LT.0.0) Y=0.0
      IF(Y.GT.YL) Y=YL
      CALL PLOT(X,Y,IPEN)
      IPEN=2
      GO TO 200
250   CONTINUE
      CALL BLOCK(XL,YL,FRQ,ZS,C0,ISF,R0,Z0,N,IHNK,ITYPEB,ITYPES,
     CRMAX,DR,WDR,WDZ,DZ,ZRR,NLYR,ZLYR,BETA,RHO,XAVG)
      CALL PLOT(XL+2.0,0.0,-3)
      CALL PLOT(0.0,-0.5,-999)
350   CONTINUE
      GO TO 100
510   CONTINUE
      CALL PLOT(0.0,-0.5,999)
      STOP
      END
      SUBROUTINE BLOCK(XL,YL,FRQ,ZS,C0,ISF,R0,Z0,N,IHNK,ITYPEB,ITYPES,
     CRMAX,DR,WDR,WDZ,DZ,ZRR,NLYR,ZLYR,BETA,RHO,XAVG)
      DIMENSION ZLYR(1),BETA(1),RHO(1)
      NC=30 ! MAX CHAR IN STRING
      HT=.1
      DY=1.5*HT
      XBLK=5.0*HT
      YBLK=YL+(21+NLYR)*DY
      CALL SYMBOL(XBLK,YBLK,HT,'IFD SOLUTION',0.,12)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'INITIAL PARAMETERS',0.0,18)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'FRQ = ',0.,6)
      CALL NUMBER(999.,YBLK,HT,FRQ,0.,1)
      CALL SYMBOL(999.,YBLK,HT,' HZ',0.,3)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'ZS = ',0.,5)
      CALL NUMBER(999.,YBLK,HT,ZS,0.,1)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'C0 = ',0.,5)
      CALL NUMBER(999.,YBLK,HT,C0,0.,1)
      CALL SYMBOL(999.,YBLK,HT,' M/SEC',0.,6)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'R0 = ',0.,5)
      CALL NUMBER(999.,YBLK,HT,R0,0.,1)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'Z0 = ',0.,5)
      CALL NUMBER(999.,YBLK,HT,Z0,0.,1)
```

```
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'N =  ',0.,5)
      FN=N
      CALL NUMBER(999.,YBLK,HT,FN,0.,-1)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'DR = ',0.,5)
      CALL NUMBER(999.,YBLK,HT,DR,0.,1)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'WDR = ',0.,6)
      CALL NUMBER(999.,YBLK,HT,WDR,0.,1)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'RMAX = ',0.,7)
      CALL NUMBER(999.,YBLK,HT,RMAX,0.,1)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'DZ = ',0.,5)
      CALL NUMBER(999.,YBLK,HT,DZ,0.,2)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'WDZ = ',0.,6)
      CALL NUMBER(999.,YBLK,HT,WDZ,0.,2)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'ISF = ',0.,6)
      FPN=ISF
      CALL NUMBER(999.,YBLK,HT,FPN,0.,-1)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'IHNK = ',0.,7)
      FPN=IHNK
      CALL NUMBER(999.,YBLK,HT,FPN,0.,-1)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'ITYPES = ',0.,9)
      FPN=ITYPES
      CALL NUMBER(999.,YBLK,HT,FPN,0.,-1)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'ITYPEB = ',0.,9)
      FPN=ITYPEB
      CALL NUMBER(999.,YBLK,HT,FPN,0.,-1)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'LYR  DEPTH(M) RHO    BETA(DB/WL)',0.,32)
      DO 600 I=1,NLYR
      YBLK=YBLK-DY
      FPN=I
      CALL NUMBER(XBLK,YBLK,HT,FPN,0.,-1)
      CALL NUMBER(XBLK+5.0*HT,YBLK,HT,ZLYR(I),0.,1)
      CALL NUMBER(XBLK+14.0*HT,YBLK,HT,RHO(I),0.,2)
      IF(BETA(I).GE.0.0)CALL NUMBER(XBLK+21.0*HT,YBLK,HT,BETA(I),0.,3)
      IF(BETA(I).LT.0.)CALL SYMBOL(XBLK+21.0*HT,YBLK,HT,'COMPUTED',0.,8)
600   CONTINUE
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'AVG =  ',0.,7)
      CALL NUMBER(999.,YBLK,HT,XAVG,0.,-1)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      YBLK=YBLK-DY
      CALL SYMBOL(XBLK,YBLK,HT,'RECEIVER DEPTH = ',0.,17)
      CALL NUMBER(999.,YBLK,HT,ZRR,0.,1)
      CALL SYMBOL(999.,YBLK,HT,' M',0.,2)
      RETURN
      END
```